

Université de Technologie de Compiègne

PhD Thesis

Multi-Sensor Perception with Vector Maps for Autonomous Vehicle Localization

Defended by: Maxime Noizet

Area of Specialization: Automatics and Robotics

Reviewers	Prof. Romuald Aufrere	Univ. Clermont-Auvergne
	Prof. Vincent Fremont	Centrale Nantes
Examiners	Dr. Joëlle Al Hage	UTC
	Dr. Mathieu Joerger	Virginia Tech
	Dr. Marie-Anne Mittet	Renault Group
	Dr. Damien Vivet	ISAE-Supaero
Supervisors:	Prof. Philippe Bonnifait	UTC
	Dr. Philippe Xu	ENSTA Paris

Heudiasyc Laboratory UMR 7253

February 4, 2025

Abstract

Multi-Sensor Perception with Vector Maps for Autonomous Vehicle Localization

For autonomous vehicles, it is crucial to find localization solutions that meet the requirements of navigation tasks. For safety reasons, the localization system must provide accurate and reliable pose estimates, with a high availability and a low latency. To this end, multi-sensor data fusion techniques are employed. They generally combine GNSS receivers with proprioceptive sensors that provide vehicle kinematics and dynamics. In this PhD thesis, we particularly focus on the use of additional exteroceptive sensors such as lidars and cameras which can provide measurements on features georeferenced in maps. These sensors help overcome GNSS limitations in complex environments like urban areas, enabling lane-level positioning.

When using perception sensors, various methods can provide localization information. A common approach in robotics is implementing Simultaneous Localization and Mapping (SLAM) with GNSS constraints. This builds a local map online from raw sensor data, which can then be used for re-localization with the same sensors. Using accurate prior maps offers an interesting alternative enabling immediate localization upon entering a new area without the need to create a map anew. In this PhD thesis, we consider high-definition (HD) vector maps containing georeferenced road features represented as points or polylines. They encompass a wide range of physical elements essential for navigation such as traffic signs or road markings.

The main goal of this thesis is to leverage all the potential offered by HD vector maps to improve localization, through a perception system whose performance has been optimized for this purpose. As a case study, our research focuses on detecting pole-like features, which are commonly found throughout road environments and georeferenced in HD maps. More specifically, we present camera and lidar perception approaches enabled by a map-based automatic annotation method. This method can annotate any kind of mapped poles. To enhance annotation accuracy and completeness, we integrate this primary method with additional automatic annotation sources. We train detectors to identify pole bases in camera images and from clusters of lidar points. This integration ensures that detected poles conform to the definitions used in the HD map. Finally, these detection approaches are integrated in a multi-sensor fusion system to assess their benefits for a localization system.

Given the approaches explored, the thesis heavily relies on experimental data collected from vehicles equipped with lidar sensors and cameras. This work was carried out in synergy with the European project ERASMO, which aimed to develop a highly accurate and reliable localization system for autonomous vehicles.

Résumé

Perception multi-capteurs avec des cartes vectorielles pour la localisation des véhicules autonomes

Pour les véhicules autonomes, il est crucial de trouver des solutions de localisation qui répondent aux exigences des tâches de navigation. Pour des raisons de sécurité, le système de localisation doit fournir des estimations de pose précises et fiables, avec une grande disponibilité et une faible latence. À cette fin, des techniques de fusion de données multi-capteurs sont employées. Elles combinent généralement des récepteurs GNSS avec des capteurs proprioceptifs qui fournissent la cinématique et la dynamique du véhicule. Dans cette thèse de doctorat, nous nous concentrons particulièrement sur l'utilisation de capteurs extéroceptifs supplémentaires tels que les lidars et les caméras qui peuvent fournir des mesures sur des caractéristiques géoréférencées dans des cartes. Ces capteurs permettent de surmonter les limites du GNSS dans des environnements complexes tels que les zones urbaines, en fournissant un positionnement au niveau des voies.

Diverses méthodes peuvent fournir des informations de localisation à partir de capteurs de perception. Une approche courante en robotique consiste à mettre en œuvre une méthode de localisation et cartographie simultanées avec des contraintes GNSS. Cette méthode construit une carte locale à partir des données brutes des capteurs, qui peut ensuite être utilisée pour la relocalisation avec ces mêmes capteurs. L'utilisation de cartes préalables précises offre une alternative intéressante permettant une localisation immédiate dès l'arrivée dans une nouvelle zone sans avoir à créer de carte. Dans cette thèse de doctorat, nous considérons des cartes vectorielles haute définition (HD) contenant des caractéristiques routières géoréférencées représentées sous forme de points ou de polylignes. Elles englobent un large éventail d'éléments physiques essentiels à la navigation, tels que les panneaux de signalisation ou les marquages routiers.

L'objectif principal de la thèse est d'exploiter tout le potentiel offert par les cartes vectorielles HD pour améliorer la localisation, grâce à un système de perception dont les performances sont optimisées à cette fin. En guise d'étude de cas, notre recherche se concentre sur la détection d'éléments de types "poteaux", que l'on trouve couramment dans les environnements routiers et qui sont géoréférencés dans les cartes HD. Plus précisément, nous présentons des approches de perception par apprentissage automatique utilisant des caméras et des données de lidar. Afin d'éviter de labelliser manuellement les données, nous étudions des méthodes d'annotation automatique qui utilisent les cartes. Ces méthodes peuvent annoter automatiquement n'importe quel type de poteau cartographié. Pour améliorer la précision des annotations, nous combinons des sources d'annotation automatique supplémentaires. Nous présentons des détecteurs pour identifier les pieds des poteaux dans les images de caméras et les poteaux dans les nuages de points lidar. Cette approche garantit que les détecteurs de poteaux sont bien adaptés aux données géoréférencées dans les cartes HD. Enfin, ces approches de détection sont intégrées dans un système de fusion multi-capteurs afin d'évaluer leurs avantages pour un système de localisation.

Compte tenu des approches explorées, la thèse s'appuie fortement sur des données expérimentales collectées à partir de véhicules équipés de capteurs lidar et de caméras. Ce travail a été fait en synergie avec le projet européen ERASMO qui visait à développer un système de localisation très précis et fiable pour les véhicules autonomes.

”Continue ta route, observe le monde de tes propres yeux et alors, peut être qu’à la fin de ton voyage tu parviendras à une conclusion différente de la nôtre.”

Silvers Rayleigh, One Piece, Eiichirō Oda

Remerciements

Voici venu le temps ~~des rires et des chants~~ de la rédaction de l'une des sections, sinon la section la plus complexe de ce manuscrit. Pour celle-ci, chacun affrontant cette étape pourrait se contenter d'une liste non exhaustive, plus ou moins ordonnée, regroupant les noms qui ont marqué sa vie, qu'ils soient d'ordre privé ou professionnel. Cette liste pourrait être enjolivée en sollicitant nos nouveaux collègues, tels que Gemini, Bard, Claude, ChatGPT, et bien d'autres encore, afin de dépasser la froideur d'une simple énumération à l'aide d'un prompt générique. Cela transformerait alors cette section en une version plus élaborée. Avec un brin de courage et une pincée d'effort, on pourrait même y intégrer des informations supplémentaires pour parvenir à un texte ayant une touche d'humanité.

Cependant, cela ne refléterait pas suffisamment l'importance de tels remerciements, après un parcours long de plus de trois ans de hauts et de bas, parfois insoutenablement long et parfois étonnamment court, passant de saisons en enfer à illuminations. C'est pourquoi nul ne peut se contenter de tels artifices, et il est ainsi nécessaire, à travers ceux-ci, de faire ressentir l'apport que chaque personne a eu dans la complétion de cette thèse (car ce n'est jamais le résultat d'une seule personne seule face au monde). Aussi, cette partie, qui doit certes rester suffisamment courte, peut être l'occasion de faire ressentir les sentiments parcourus pendant cette thèse ~~ainsi que la longueur de celle-ci~~.

Voici donc maintenant plusieurs semaines que je me retrouve face à cette page ~~anciennement~~ blanche, tel un nouveau processus créatif, à ne pas savoir comment tourner ceux-ci pour exprimer au mieux ma gratitude. Plusieurs personnes remerciées par la suite pourront témoigner de ce processus créatif. J'ai d'abord pensé à faire des remerciements relativement conventionnels, mais je trouve ça assez déplaisant et très ressemblant à l'idée émise au début de cette section. J'ai ensuite pensé à enchaîner les remarques ~~essayant d'être~~ drôles, mais je pense qu'il n'est pas encore temps d'écrire mon one-man-show. J'ai ensuite tenté d'écrire un discours, posé, réfléchi, travaillé et de bien choisir le moindre mot, mais cela manquerait à mon sens de spontanéité. En effet, la spontanéité est très importante puisqu'elle est synonyme de remerciements honnêtes. Par ailleurs, la spontanéité est en quelque sorte au cœur d'une thèse, c'est avec spontanéité qu'arrivent les idées, bonnes comme ~~souvent~~ mauvaises, et c'est ensuite qu'il faut faire le tri.

Ainsi, pour que mes remerciements soient à la hauteur du ressenti que je souhaite provoquer chez vous, lecteurs, j'ai décidé d'abandonner les artifices habituels et d'écrire spontanément, et cette spontanéité formera la version finale de mes remerciements ~~qui promis vont commencer juste après~~, néanmoins relue et corrigée légèrement, car bon, il ne faut pas abuser non plus.

CEPENDANT! Il ne faudrait pas oublier… Blague à part, commençons.

Tout d'abord, et parce que c'est notre projet ! Je tiens à exprimer ma gratitude envers Philippe et Philippe pour leur encadrement dans le cadre de cette thèse et du projet européen associé. M'avoir offert cette opportunité m'a permis d'effectuer un ~~grand~~ pas dans le monde de la recherche et d'explorer mes ~~modestes~~ idées pendant ces quelques années où, mine de rien, nous avons partagé de nombreuses heures d'échanges et de débats d'idées, de collaborations scientifiques, mais également techniques, qui m'ont permis d'aiguiser mes compétences sur les sujets traités dans cette thèse. (Philippe)^2, vous avez su être à l'écoute et me prodiguer vos connaissances et vos conseils, malgré vos emplois du temps, que l'on sait tous, chargés. Je pense qu'ensemble, nous avons su développer une certaine synergie qui a permis d'arriver

aux travaux que je suis fier de présenter aujourd'hui dans ce manuscrit. Je n'aurais jamais imaginé arriver aux résultats auxquels nous sommes parvenus.

Je tiens à remercier les rapporteurs et membres du jury, car ce sont eux qui ont permis de mettre un terme à cet épisode de ma vie extrêmement positivement. Je tiens à vous remercier particulièrement pour vos lectures attentives de ce manuscrit, vos retours et nos discussions extrêmement riches et enrichissantes dans le cadre de ma soutenance. Celles-ci, par leur pertinence, seront essentielles à prendre en compte dans le cadre de travaux supplémentaires sur ce sujet.

Je tiens à remercier l'ensemble des collègues du laboratoire pour les moments que nous avons passés ensemble ces dernières années, la plupart du temps autour d'un café qui avait parfois tendance à s'éterniser, voire à refroidir, à rire, discuter, mais très souvent à nous plaindre de tout et de rien, et à parler science, car oui, parfois ça arrivait.

Je tiens à remercier l'ensemble du personnel administratif, sans qui nous serions très souvent perdus dans toutes les démarches que nous pouvons être amenés à effectuer. Par ailleurs, à ce sujet, je vous remercie pour l'aide apportée lors des divers enchaînements de contrats différents qui ont pu avoir lieu, ainsi que pour toutes les missions réalisées, presque sans accroc. Pour cela, on ne remerciera bien évidemment pas Notilus.

Je tiens à remercier l'ensemble des permanents et doctorants, particulièrement ceux de Sivalab, pour nos nombreux échanges et la mise en commun de nos solutions techniques, qui ont tendance à nous faire économiser un temps fou, ainsi que pour l'aide apportée pendant le déroulement du projet. Pour cela, je remercie particulièrement Antoine, Maxime, Rémy, Corentin, Stéphane, Thierry et Joëlle pour leur aide à leur échelle. Par ailleurs, Stéphane, Thierry, merci pour toutes les fois où vous avez répondu à mes sollicitations, interrogations, et j'en passe. Sans vous, nous serions incapables d'utiliser cette plateforme dont nous avons la chance de pouvoir profiter, et bon nombre de travaux ne seraient pas ce qu'ils sont sans cela. Merci Stéphane pour les nombreuses aides techniques, qui dépassent même le cadre de la plateforme.

Jean-Benoist, je te remercie particulièrement pour l'année ~~plus longue que prévue~~ précédant la thèse, durant laquelle, avec Philippe, j'ai pu travailler sur des problèmes de recherche communs. J'ai énormément appris en travaillant avec toi, même si, malheureusement, j'ai l'impression d'avoir oublié tant de choses. Même pendant ma thèse, j'ai toujours eu l'occasion d'apprendre de ton expertise, notamment en enseignant SY02 avec toi. Aussi, j'admire ta capacité, partagée avec Stéphane, à avoir un flot de connaissances aussi vaste sur tout et ~~surtout~~ n'importe quoi, et à nous absorber à ce point dans des conversations souvent très longues.

Joëlle, merci pour toutes nos discussions, scientifiques ou non. Nous partageons de nombreuses thématiques de recherche, et nos échanges, que ce soit pendant les CSI, la soutenance ou à d'autres occasions, ont toujours été très enrichissants. Tu as souvent su m'aider, dans les phases les plus complexes de la thèse, et pour cela, je te remercie.

En évoquant ces phases difficiles, il me semble important de souligner combien j'ai sous-estimé, comme beaucoup de doctorants, la traversée psychologique qu'implique un doctorat. Je pense que, s'il y a bien un sujet sur lequel on apprend pendant le doctorat, c'est sur soi-même. Cette traversée peut parfois se faire dans la douleur, mais elle devient généralement beaucoup plus agréable grâce aux personnes présentes pour nous aider à la surmonter.

Pour cela, je remercie tout d'abord le groupe des imposteurs : Soundouss, Loïc, Nicolas, Sana, Lahcene. Nos discussions, les jeux pendant les pauses et en dehors, ont fait de cette

période, avec du recul, un moment très agréable à vivre. Je pense que sans vous, le bilan aurait très certainement été plus mitigé. Il y a dans la vie des rencontres marquantes et d'autres moins. Je vous considère parmi les premières et je suis content que cette thèse ait été l'occasion pour cela, sauf pour Loïc, non pas parce que je ne t'apprécie pas, mais parce que l'occasion s'est présentée bien avant, dès le premier jour, voire la première minute, de notre entrée à l'UTC. Parmi ces rencontres marquantes, il y a une personne qui se considère aussi comme un imposteur parfois : merci Lyes pour ton soutien ces dernières années. Et merci aussi de m'avoir motivé à me remettre au sport pour évacuer le stress. Cela aura eu le double bénéfice de réduire mon stress et de me remettre en forme.

Merci à Jean-Benoist, Joëlle, Sabine, Sylvain, Sébastien, Rémy, Antoine, Maxime, Corentin, Philippe, Jean-Paul, Vitor, Michaël, pour les nombreuses discussions, notamment en salle café, d'avoir partagé mon humour, ou peut-être subi et d'avoir fait semblant de l'apprécier.

Merci à mes amis de longue date, avec qui j'ai partagé ma scolarité à l'UTC : Célien, Constance, Emilien (pour une fois que j'utilise ton prénom), Elias, Eumaël, Stéphane, Vincent. Vous avez tous supporté mon humour et mes travers pendant de nombreuses années, et pour cela, je vous remercie. Même si nous nous voyons moins souvent, je suis content que, malgré le temps, nous passions encore de bons moments ensemble et puissions toujours discuter, peu importe le moment. Pour certains, nous avons partagé un Hellfest mémorable, mais un peu "covidé" pour ma part. Désolé Eumaël pour la nuit horrible et ce partage de virus ~~dans une si petite tente~~. Merci Eumaël et Stéphane pour votre immense soutien ces dernières années. Vous avez tous les deux parfois subi mon comportement. Évidemment, je ne pensais pas certains mots que j'ai employés, et je vous confirme l'amitié réelle derrière ces faux-semblants. Par ailleurs, Stéphane, c'est en partie grâce à toi si j'ai pu vivre cette thèse. Ton soutien face à mon stress et mes doutes a été essentiel, et je te remercie d'avoir, à ton échelle, permis cela.

J'ai partagé avec vous tous des moments parmi les plus joyeux de ma vie et je vous en remercie et j'espère encore en partager de nombreux. Je m'excuse d'avance pour tous les noms que j'ai pu oublier jusqu'à maintenant, je remercie tous ceux qui ont pu être présent à un moment ou à un autre dans ma vie, même si nos chemins se sont séparés pour X raisons.

Enfin, je ne pouvais terminer ces remerciements sans évoquer ceux sans qui je ne suis finalement vraiment pas grand-chose. Ceux pour qui je souhaiterais exprimer tant de choses, tellement ma vie ne serait pas la même sans leur existence. Étrangement, ce sont pour ces personnes que les mots sont les plus difficiles à écrire, alors qu'ils m'évoquent pourtant tout un tas de sentiments et d'événements positifs quand je pense à eux.

Lucas, je te remercie pour notre ~~(trop ?)~~ longue amitié et ton soutien psychologique permanent, particulièrement pendant ces dernières années, où, on ne va pas se mentir, je n'ai pas toujours été facile à vivre. Sana, je te remercie pour la personne formidable et l'incroyable partenaire ~~commercial~~ que tu es, pour le soutien que tu m'apportes au quotidien. Vous êtes certainement tous les deux ceux qui ont le plus subi mes différents états d'âme, et pour cela, je pense que je n'aurai jamais assez de mots ni de temps pour vous remercier.

Pour finir, Maman, Papa, sans vous, ne serait-ce que biologiquement parlant, je ne serais pas là aujourd'hui. Sans votre éducation, votre soutien, vos sacrifices parfois, je ne serais pas le docteur que je suis devenu aujourd'hui.

Contents

List of Figures

List of Tables

List of Abbreviations

AD	Autonomous Driving
ADAS	Advanced Driver Assistance System
AP	Average Precision
CCDA	Combined Constraint Data Association
CNN	Convolutional Neural Network
ECEF	Earth-Centered, Earth-Fixed
ENU	East-North-Up
(E)KF	(Extended) Kalman Filter
FN	False Positive
FP	False Negative
GNSS	Global Navigation Satellite System
HD Map	High Definition map
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IoU	Intersection over Union
LC	Loosely-coupled
MAE-x	Mean Absolute Error on horizontal axis
NMS	Non-Maximum Suppression
ODD	Operational Design Domain
OSM	Open Street Map
PCA	Principal Component Analysis
POI	Point Of Interest
PPP	Precise Point Positioning
PPK	Post-Processed Kinematic
PR	Precision-Recall
RTK	Real-Time Kinematics
S/GBAS	Satellite-/Ground-Based Augmentation System
SLAM	Simultaneous Localization and Mapping
SPP	Single Point Positioning
TC	Tightly-coupled
TP	True Positive
(U)NN	(Unique) Nearest Neighbor

xxx

List of Symbols

$\mathcal{A}$	Set of annotations.
${}^{(k)}a_j^i, {}^{(k)}a^i, {}^{(k)}a$	Annotation j, on image i, and across all images using method k.
${}^{(1:K)}_{q}a^i$	Consensus annotation set where at least q methods agree.
$B^i = \left\{B_j^i\right\}_{j=1,\dots}$	Bounding boxes from annotations on image i.
${}^{L}\mathcal{C} = \left\{\mathcal{C}_j\right\}_{j=1,\dots}$	Clusters from point cloud in frame L.
$D^i = \left\{D_j^i\right\}_{j=1,\dots}$	Detected bounding boxes on image i.
$\mathcal{D}_{k,i,j}$	Distance between objects i and j at time k if specified.
${}^{A}\mathcal{G}, {}^{A}\overline{\mathcal{G}}$	Ground/non-ground points in lidar cloud in frame A.
${}^{A}\mathcal{G}_i, {}^{A}\overline{\mathcal{G}}_i$	Ground/non-ground points near pole i in frame A.
${}^{A}\mathcal{M}$	Pole map in frame A.
${}^{A}\mathcal{M}_D$	Map poles near the vehicle in frame A.
${}^{A}\mathcal{M}_D^*$	Corrected map poles near the vehicle in frame A.
${}^{C}\mathcal{M}_k^\alpha$	Pole map in frame C as bearings at time k.
${}^{A}\mathbf{m}_{k,j}^\alpha$	Map feature j in frame A at time k, transformed to bearing if α.
${}^{A}p_i$	Point i in frame A.
${}^{A}P_i$	Map pole i in frame A.
${}^{A}\mathcal{P}$	3D lidar point cloud in frame A.
${}^{(k)}r, {}^{(k)}p$	Recall/Precision of method k.
${}^{(1:K)}_{q}r, {}^{(1:K)}_{q}p$	Recall/Precision of consensus set.
S	Image segmentation mask.
${}^{B}T_A$	Transformation matrix from A to B frame.
$\mathbf{x}_k = \left[x_{B,k}, y_{B,k}, \theta_{B,k}, v_k, \dot{\theta}_k\right]^\top$	Vehicle state vector in frame O at time k.
${}^{A}\mathbf{Y}_k^B$	Type B detections in frame A at time k.
$\boldsymbol{z}_k^B$	Type B observation at time k.
$\alpha_{k,i}$	Bearing of object i at time k.
$\boldsymbol{\beta}_{k,i}^B$	Observation noise for object i, type B at time k.

Localization for autonomous driving

Contents

Introduction

In Autonomous Driving (AD), obtaining a sufficiently accurate localization solution in line with the needs of the navigation task is essential. In fact, depending on the context and its needs the localization requirements can be challenging to meet as, for example, in urban environment to stop at an intersection or navigate on narrow roads. To obtain an accurate localization anywhere in the world, Global Navigation Satellite Systems (GNSS) are commonly used. Using GNSS receivers, estimated poses can be computed by estimating the distances between satellites and receivers through satellite signals.

A meter accuracy is generally reached using GNSS alone in open sky. GNSS performance often degrades in environments characterized by multipath and Non-Line-Of-Sight (NLOS) signals, making them unsuitable alone for these applications. Even on highways, where GNSS-based localization seems promising, it may be insufficient, since lane-level positioning is generally needed for Advanced Driver Assistance Systems (ADAS) and AD leading to strict requirements. Relying solely on GNSS and proprioceptive sensors (including wheel speeds and Inertial Measurements Units (IMU) proves insufficient in such cases, necessitating the integration of more robust localization techniques.

To improve localization performance, data from multiple exteroceptive sensors, such as lidars or cameras, can be fused with GNSS and proprioceptive sensors. Information from maps can also be added to exploit perception data. Different types of maps provide diverse insights into the road environment. Perceiving the environment with different sensors and building a representation consistent

with the maps used can help provide additional information on the current vehicle pose.

Besides, as being safety-critical, autonomous vehicles must have high confidence in their localization estimates to avoid any dangerous situation. To this purpose, a measure of trustworthiness called integrity plays a crucial role. Unlike accuracy, defining integrity standards is complex for AD. Even if standards are well-defined in other fields such as aviation, there is no consensus for intelligent vehicle localization. Analyzing the integrity of localization estimates ensures reliability. However, as localization involves multiple data sources, assessing the trustworthiness of each source seems vital to ensure reliable localization estimates. Using faulty or untrustworthy data in the estimation process without excluding them can lead to significant localization errors and integrity issues. Quality monitoring procedures for the measurements used must be established to achieve the targeted performance. This is particularly the case when data contained in a map and perception data from the environment are brought together.

This thesis aims to enhance the localization system of autonomous vehicles, particularly in terms of accuracy. Acknowledging the complexity of the task, which involves multiple subproblems, we chose to focus on managing environmental perception, within a specific mapped operational domain, dedicated to localization.

THE ERASMO PROJECT

Providing a high-accuracy and high-integrity localization solution was the main objective of the European project ERASMO (Enhanced Receiver for Autonomous MObility). The project ran from July 2021 to June 2024. To reach this goal, several partners were brought together: Idneo, GMV-AD, GMV-ITS, Heudiasyc CNRS/UTC, Artisense, Renault and Septentrio. Several technological challenges spread across the different partners were overcome such as:

- The optimization of GNSS signals especially Galileo constellation signals in terms of accuracy, availability and integrity, through the use of signal corrections.
- The use of exteroceptive and proprioceptive sensors to provide complementary information for the pose estimation.
- The fusion of all sources of localization information and estimation of integrity to comply with project requirements.

The simplified architecture of the system designed in this project is shown in Figure 1. A main filter is responsible for providing a high-accuracy estimation with high integrity by fusing GNSS/IMU data with two external exteroceptive sources.

One of these sources is the Artisense Visual Inertial Navigation System (VINS) which uses an IMU and stereo cameras. Their solution is based on Simultaneous

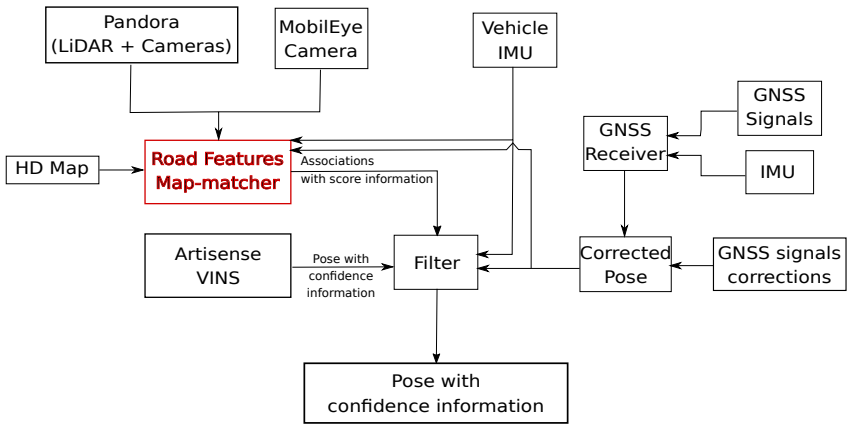


Figure 1: *Simplified architecture of the ERASMO project. In red, the module that Heudiasyc was responsible for.*

Localization and Mapping (SLAM) and relocalization capabilities. A map is built using visual features extracted from images then a pose is provided in real-time by identifying correspondences between map images and the newly acquired one.

The second one from Heudiasyc was the design of a module that matches detections from cameras and lidar sensors with landmarks stored in a vector map. This thesis contributions are carried out in this context. Having such a diverse range of localization information sources enhances the robustness of the entire system and significantly improves its availability.

PROBLEM STATEMENT AND OBJECTIVES

As previously introduced, when trying to reach the localization performance necessary for AD, perception data are inevitable when using maps. Despite the ideal scenario where autonomous vehicles operate in all environments with optimal performance, the reality is far more intricate. Consequently, they are typically engineered to function within a specifically defined Operational Design Domain (ODD). This domain encapsulates all navigation conditions the vehicle must comply with, including road context, weather conditions. It defines the necessary performance across various operational scenarios to guarantee safety, notably in terms of localization.

In order to ensure the availability of the autonomous system, the ODD must be respected. It is always desirable for this ODD to be as broad as possible, taking into account the architecture of the available autonomous vehicle.

To achieve this goal and to address as many scenarios as possible, sensor diversity and redundancy are essential. As mentioned earlier, GNSS can encounter limitations in certain conditions and must be supplemented with other sensors such as lidars and cameras. However, these sensors can also face challenges in specific circumstances. Cameras are typically sensitive to varying illumination conditions, while lidars may be affected by weather conditions.

Nowadays, the architectures of autonomous vehicles vary widely. Indeed, no standard architecture or specific sensor requirements have been defined yet, even for localization. Although the most commonly used sensors today include GNSS, IMUs, cameras, and lidars in a complete system, there is a great diversity among these sensors. Thus, like the ERASMO system, localization systems are generally seen as a fusion module that takes inputs from several 'black boxes' providing positioning information to help estimation, each with an unknown process for producing them.

Therefore, given the diverse and modular nature of autonomous vehicle architectures, equipped with multiple perception sensors to ensure functionality in different conditions, it is necessary to propose localization methods that are sensor-agnostic as much as possible. This is particularly the case for methods that use vector maps for localization. Indeed, these maps, contain geometric information about the environment, including the positions of certain features. Detecting these features enables their geometric association with corresponding elements in the map and consequently provide localization information.

However, when relying on a vector map to integrate perception data, it can occasionally be problematic to deliver usable localization information. This is particularly evident in areas where the map lacks adequate detail compared to perception data, or conversely, when perception data fails to provide sufficient information compared to the map. Furthermore, it can be challenging to accurately associate detections with map elements in difficult contexts, and maps can be prone to errors.

To address the aforementioned problems, the visual detection should align as closely as possible with the map. It means that efficient detection of as many elements as possible from the available map, while avoiding detection of objects not covered by the map's semantics, is needed. Additionally, to fully leverage the sensor-agnostic nature of vector maps and provide extensive localization data to enhance overall performance, especially availability, these detectors must be developed for widely used sensors such as cameras and lidars. The proposed detectors should be adaptable to any context or experimental setup without requiring specific additional human effort, ensuring compatibility with various architectures using vector maps for localization.

The entire system also needs to satisfy several requirements. Adequate information availability at all times necessitates real-time operation of road features detection, respecting the sensor data acquisition rate. Similarly, the association of detected objects with mapped features must meet a similar real-time constraint to provide localization data at an acceptable frequency for localization. In fact, given the critical role of localization in various components of intelligent vehicles, such as control, it must satisfy temporal constraints. For control, a high-frequency estimation is particularly needed and estimation needs to be done with little latency. Finally, the entire system must respect the accuracy requirements of the navigation task.

THESIS CONTRIBUTIONS

The main scientific contributions of this thesis are:

- A map-based automatic pole base annotation approach for cameras [Missaoui et al., 2023] and lidars: Maps store information about pole-like road features in the environment. By collecting data from autonomous vehicles drives in real driving contexts, images can be annotated with map data to build a pole detector afterward. Data acquired by the vehicle is used to compute a post-processed reference pose used to project map data onto data taken by the onboard sensors. This allows building detectors with limited human effort, for any road condition, as soon as there is an available map.

- Multi-modal automatic annotation for cameras [Noizet et al., 2024] and lidars: By applying post-processing methods similar to those used with map data, other annotation approaches can be used. Non-real-time algorithms are employed to detect poles in the acquired data and generate annotations. These algorithms enable the direct extraction of poles from images and lidar point clouds. In images, annotations from lidar point cloud segmentation, map and image segmentation can be obtained. For a lidar, point cloud segmentation and the map can be used. For both sensors, multiple sets of annotations are merged to build more accurate training sets.

- Pole detectors for cameras and lidars: Automatic annotations obtained are used for pole detectors training. For images, classical deep learning based object detection methods are adapted to propose such a detector with a data representation usable in our context and consequently consistent with our maps. For lidar, clustering classification methods are developed.

- A multi-sensor localization system mainly based on poles detected by lidar and cameras and associated with data contained in maps [Noizet et al., 2023]: An analysis of the impact on the global positioning in terms of accuracy is proposed. The gain proposed by a multi-camera system integrated to a GNSS and Dead Reckoning (DR) sensors is evaluated. A similar evaluation by replacing the multi-camera system by a lidar is also conducted.

- An experimental evaluation of the proposed perception and localization methods: These methods are tested on real data recorded with an experimental vehicle on open roads under real driving contexts. All the sensors used are expected sensors for the developed methods and therefore particularly suitable for a highly autonomous vehicle.

As part of the European ERASMO project, several contributions beyond this thesis have been made. These include participation in the development and integration of the ERASMO solution, data acquisition, preparation for its demonstration and evaluation, and involvement in the dissemination activities.

MANUSCRIPT ORGANIZATION

The manuscript is composed of the following chapters that are briefly summarized here:

- Chapter 1: introduces the navigation environment studied and the importance of maps for navigation and localization is detailed. Studying different types of maps provides insights into the types of information that are usable for localization. Our case study focuses on improving localization by adding sensor-agnostic vector maps for positioning relative to map data. We describe the maps used in this context and detail how the information about road features contained are used and the potential arising difficulties. We detail state-of-the-art methods to extract road features from lidar and cameras using geometric approaches that could be used with sensor-agnostic maps and highlight how machine learning and deep learning approaches have surged in importance for perception. Then, we introduce the problem of localization with mapped landmarks, and particularly the problem of associating mapped features and detections.

- Chapter 2: details state-of-the-art methods for automatic annotation of datasets when there is a lack of data to build detectors. We emphasize the necessity of automatic annotation to reduce the current human effort required. To improve the availability of the perception system for localization during an entire driving, i.e. to provide regular detections of road features, we choose to detect all pole-like features stored in the map. Here, we introduce automatic annotation approaches designed to develop detectors for cameras, using data from maps, lidar point clouds, and images. The primary emphasis is on enhancing automatic annotations derived from map data, as the vector map plays a central role, to guarantee that the built detectors respect the map's definition. A multi-modal automatic annotation approach is derived, adapted for object detection.

- Chapter 3: details state-of-the-art methods for object detection using neural networks, including uncertainty management in the training set and performance evaluation techniques. Then, it introduces the trained detectors using the automatically annotated images previously introduced. To avoid developing new neural networks tailored to this specific context, we choose instead to adapt widely adopted object detection networks to obtain a pole base detector that meets real-time constraints and thus can be applied for localization. We firstly present the performance of detectors trained only on map-based annotations. Then, we study how to enhance the overall performance by integrating other annotation sources. We propose a method to manage annotation uncertainty by masking elements in the image when there is doubt if they correspond to poles. As the obtained detectors are sensitive to annotation errors and their final performance is limited by them,

we conduct an initial study on the impact of annotation errors on the training. As we want to limit the human effort when constructing detectors, we provide first insights on the need of manual annotations for validation.

- Chapter 4: introduces various GNSS solutions that can be used in a multi-sensor system and describes the data association techniques used to infer localization information from detected objects by matching them with their corresponding mapped features. The data fusion strategies in such a multi-sensor system are also introduced. Then, it presents the approach developed to provide localization information using maps and our pole base detectors in a complete system. We analyze the contribution of the developed pole base detectors to localization accuracy, while highlighting the challenges encountered in a multi-sensor localization system largely based on sensor-agnostic maps. For that, we analyze the contribution with different GNSS sources providing different levels of accuracy.

- Chapter 5: After having demonstrated the feasibility of training camera-based pole base detectors without existing datasets and by minimizing human effort through multi-modal automatic annotation, we further validate that our approach can be adapted to other sensors. We conduct a similar study for lidars. We present a multi-modal annotation method for lidar clusters using vector maps and lidar segmentation network. Following this, we introduce a classification method for lidar clusters based on geometric rule learning. Finally, we integrate the developed detector into a multi-sensor system to assess its impact on localization accuracy.

Chapter 1

HD Maps: a backbone for autonomous navigation and localization

Contents

1.1 Navigation environment

The advantages of autonomous vehicles are manifold. If their deployment could be widespread, it would enable universal access to transportation, even in less accessible areas, while potentially reducing accidents and traffic congestion. However, the widespread implementation of such vehicles appears unfeasible, largely due to the high costs associated with current architectures and the limitations of current technology. Nonetheless, their development paves the way for new forms of public transportation with the emergence of autonomous shuttles, like those developed by companies like Zoox or EasyMile.

Such shuttles help diversify and enhance the existing public transportation offerings, easing travel between key locations that may not be currently served. In a typical mid-sized French city, they could make it easy to connect industrial and commercial areas with the city center. An example of a route is shown in Figure 1.1, covering a wide range of environments, from open highway areas to dense urban zones.

Figure 1.1: *Satellite View of roads navigable by an autonomous vehicle (in black)*

In fact, during navigation, the shuttle may potentially drive in various environments, as depicted in Figure 1.2. Therefore, the system must achieve optimal performance in these diverse environments to ensure autonomous navigation, particularly in terms of localization.

It is evident that a multi-sensor approach is necessary to navigate under these conditions. In environments depicted in Figures 1.2c and 1.2d, GNSS performance is lower. Even in environments like the one shown in Figure 1.2b, GNSS performance can be degraded, especially near bridges.

This justifies adding perception sensors to localize the vehicle relatively to the observed environment. Cameras and lidars are particularly suitable to help localization in environments similar to those in Figures 1.2c and 1.2d due to the density of visible information. Lidars are particularly useful. These sensors scan the environment using lasers, and for 360-degree lidars, a mechanical rotating platform enable generating regular 3D point clouds. These point clouds capture the environment more or less densely, depending on the horizontal and vertical angular resolution of the sensor. In such environments, this provides a substantial amount of information about the surroundings, aiding in recognition and consequently, localization.

Besides, the vehicle must be able to navigate under various conditions, including weather and traffic. Sensors may be more or less affected by these conditions, justifying mixing different perception modalities.

(a) *Open sky.*

(b) *Highways.*

(c) *Urban area.*

(d) *Peri-urban.*

Figure 1.2: *Typical driving environments.*

Figure 1.3: *Common road features in navigation environment*

Road environments can be identified by the visible features they contain. Indeed, road features such as traffic signs, traffic lights, streetlamps, and even bollards, as seen in Figure 1.3, line the various roads and can serve as visual landmarks for accurate localization within a lane or along a road.

As explained earlier, providing an autonomous vehicle that operates anywhere is complex. Typically, it operates within a specific geographical area, so it is not critical if it does not work outside this area. However, it should be adaptable to a potential new driving environment with minimal effort. For any task related to autonomous navigation, especially for localization, it is possible, even essential,

to develop specialized approaches for the geographical area navigated by the vehicle. For localization purposes, this entails deploying perception methods specifically designed for these regions, finely tuned to the geographical area under exploration. The use of maps creates a link between what is detected at a given moment and a landmark in the geographical environment.

1.2 MAPS AND SLAM-BASED RELATIVE POSITIONING

1.2.1 *From standard to high-definition maps*

Nowadays most of the maps used are digital maps mainly used to simplify navigation for road users through applications such as Maps or Waze, called turn-by-turn navigation. Even digital maps are stored in vehicles for navigation or for ADAS systems. Typically, the first digital maps available were maps called "standard maps" as Google Maps or Open Street Maps (OSM), an equivalent free and open collaborative map database commonly used for data visualization and turn-by-turn navigation. These maps are generally built through surveys and data extraction from aerial and satellite images making them sufficiently accurate for the aforementioned task. They generally encode road structures and basic semantic information as well as points of interest (POI). To be usable, a topological layer is generally accessible providing connectivity between the roads. The information is generally limited with a low accuracy of few meters. Particularly, depending on the methods used for their design, the available information and their accuracy can vary.

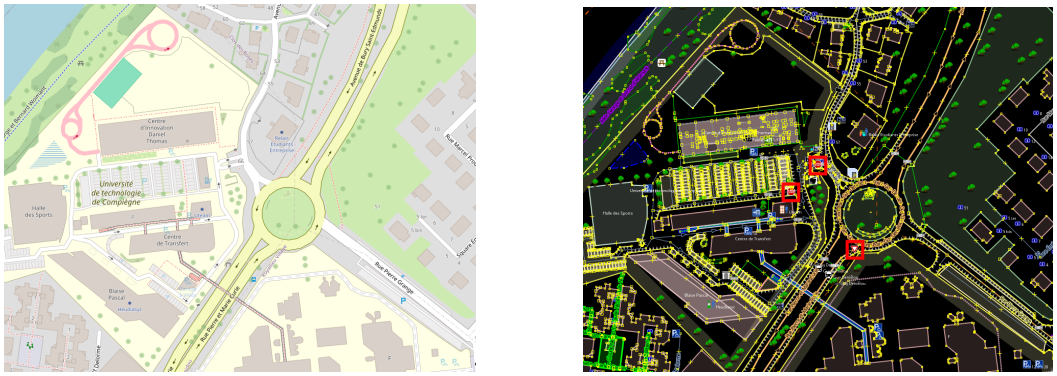
Digital maps contain road-level information without any data on the lanes composing it. Usually, when using turn-by-turn navigation applications, no information about exact lane positioning is provided. The number of lanes is often used to indicate which lane to take for the desired direction. However, the application cannot recognize the current lane, potentially leading users to make mistakes and take the wrong direction despite the instructions.

When using the OSM format, a large amount of data can be encoded, including roads, buildings, road edges and markings, and even road signs. An example of this extracted database from a city in Finland is shown in Figure 1.4. These signs were mapped by contributors using videos and numerous photos.

All these elements could be used for navigation, and localization if accuracy was assured. However, accuracy remains quite low, and the information, varying by area and contributors, may be very limited. This is evident in Figure 1.5 depicting the OSM around our laboratory where only three signs are mapped.

To extend their application and make them usable for ADAS systems, digital maps were enhanced with lane-level information, speed limits, traffic signs and lights, with a higher accuracy. However, as explained by [Elghazaly et al., 2023], these maps, while more detailed, are still limited in terms of data available and accuracy for AD systems. That is why, High Definition (HD) maps were introduced

Figure 1.4: *Extract of Open Street Map with traffic signs in Finland. The traffic signs were surveyed using videos and series of shooting for large scale mapping (Wikipedia).*



(a) *Classical view.* ***(b)*** *Database.*

Figure 1.5: *Open Street Map around the laboratory (April 2024). Within the database, only three signs are accessible across this area as highlighted with red rectangles.*

in the 2010s by Daimler[1]. As explained by [Elghazaly et al., 2023], these HD maps can be particularly dense and contain extremely varied information to assist with all tasks related to autonomous vehicles.

These maps propose a high level of information containing a detailed geometric representation of the roads and environments, detailed semantics and a centimeter-level accuracy enhancing the capabilities of an autonomous vehicle in terms of navigation. Currently, there is no guidelines or standard for HD maps definition [Ebrahimi Soorchaei et al., 2022]. However, generally they are organized under multiple layers whose quantity varies depending on the utility of the HD map, its specific requirements, the manufacturer, ... [Ziegler et al., 2014; Aeberhard et al., 2015].

While we can have layers providing static information for autonomous navigation, considering the complexity of the task, it may be beneficial to have more real-time data, which can be incorporated into the definition of certain HD maps after their creation. [Elghazaly et al., 2023] introduced six generic layers for HD maps:

- The base map containing representation built from raw data from sensors as point clouds or raster images or voxel map. This layer corresponds to maps generally called dense maps.
- The geometric map containing high-precision lane-level geometric primitives (points, polylines, polygons). It corresponds to a high-definition version of vector maps.
- The semantic map containing semantic information about road features (traffic lights, road signs, pedestrian, crossing, POIs).
- The road connectivity describing map topology, i.e. how geometric primitives of the geometric layer are connected.
- The priors map, derived from past experiences, capturing evolving information over time, particularly regarding geometric and semantic elements. This layer also captures data to help predict human driving behaviors and dynamic traffic light states at intersections.
- The real-time map, a dynamic layer, providing real-time details about the environment, including traffic conditions, road closures, and other relevant events affecting autonomous vehicle navigation. Information is collected in real-time and feeds into the HD map, sourced either from participating vehicles through crowdsourcing or from intelligent infrastructure.

The first four layers are basic elements of an HD map, while the others are more advanced layers requiring sophisticated infrastructure and substantial resources for their creation. For localization, the usable layers are the base map, geometric

[1] https://www.here.com/learn/blog/the-evolution-of-the-hd-live-map

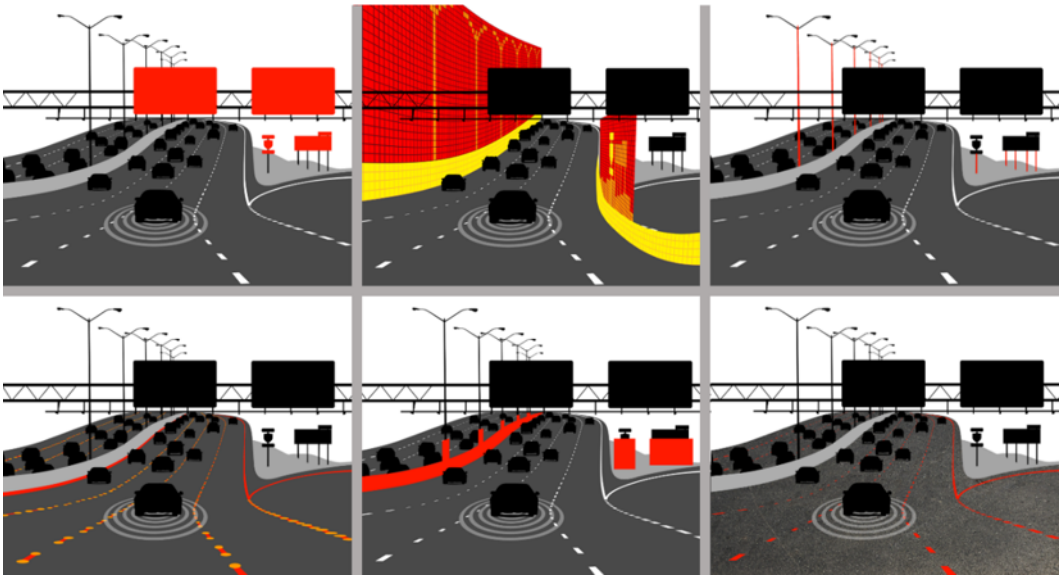


Figure 1.6: *Elements contained in Tomtom HD Map RoadDNA: from top-left to right-down, a collection of traffic signs, optimized lidar point clouds of roadsides, poles, lane markings, radar data of roadway objects and reflectivity of road surfaces extracted from lidars are provided into multiple layers. This map is self-localization oriented.*

map and the semantic map. Depending on the layer used, the approaches differ with their own advantages and drawbacks.

Due to the lack of a universal definition, some researchers consider any map with a 3D world representation as an HD Map. For some researchers it can correspond only to the base map layer or to the geometric layer, without the complexity established with the six layers defined.

For example, TomTom proposes the RoadDNA[2], an HD map containing multiple layers of information particularly oriented for sensor-agnostic localization. These layers are visible in Figure 1.6 and contains sensor-agnostic information as poles, traffic signs or road markings position but also sensor-oriented data as road reflectivity or optimized lidar point clouds of roadsides.

HERE also proposes its own HD Map[3], as visible in Figure 1.7, composed of three layers. One of them is dedicated to localization and provides the positions of road markings and roadside furniture. Here, the three layers correspond to a blend of geometric and semantic layers with road connectivity.

The HERE HD map is updated in near-real time through crowdsourcing with sensor data from connected vehicles, corresponding to a part of the priors map layer previously introduced. Furthermore, a partnership between HERE and Mobileye[4] has been established to enhance the update capability of the HERE map. Mobileye, a manufacturer of smart cameras, also provides HD maps, created and

[2] https://www.tomtom.com/products/hd-map
[3] https://www.here.com/platform/HD-live-map
[4] https://www.here.com/learn/blog/here-and-mobileye-crowd-sourced-hd-mapping-for-autonomous-cars

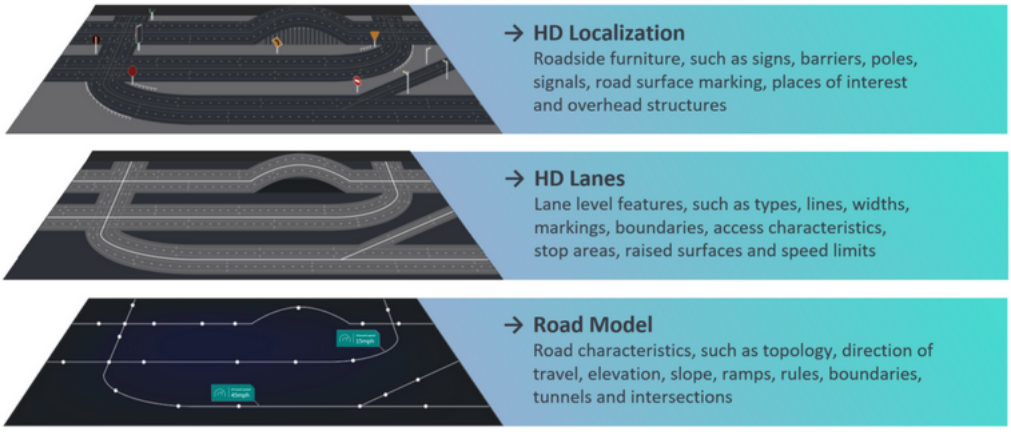


Figure 1.7: *HERE HD Map composed of three layers*

updated from information gathered through their cameras installed in a large portion of the global automotive fleet.

As seen with these examples, the definition and even the generation process of HD maps can differ a lot [Bao et al., 2022]. This lack of standard can be a huge drawback of using HD maps in an autonomous vehicle. This leads to a solution that is necessarily tailored to a specific type of HD maps, rather than being universally suitable for all. Depending on the vehicle used, the selected HD map has to be the most suitable. Therefore, until standards are defined, it is essential to choose the map consistently with the desired application, objectives, and system expectations. This applies equally to localization. It is therefore necessary to study different approaches for using perception data to localize using maps of all types, to assess their advantages and disadvantages in order to justify a given choice.

Furthermore, even with an extremely accurate map of the environment, localization with it can prove to be complex, regardless of the format. Indeed, [Javanmardi et al., 2018] explained that prerequisites are necessary for localization and thus defined four criteria applicable to any map. Additionally, they defined specific metrics to use for a specific dense map type. These criteria are features sufficiency, the layout of the map, the local similarity of the map and the representation quality. A sufficient amount of high-quality features should be stored in the HD map evenly distributed in the navigation environment and local similarity should be avoided to avoid map ambiguities. Depending on the type of the map, it is difficult to avoid some of these issues since they mainly depend on the level of abstraction and the environment.

1.2.2 *Dense maps*

As previously introduced, HD maps can contain a layer representing raw sensor data. The dense maps can be of multiple types, from the original acquired data to an abstraction of them.

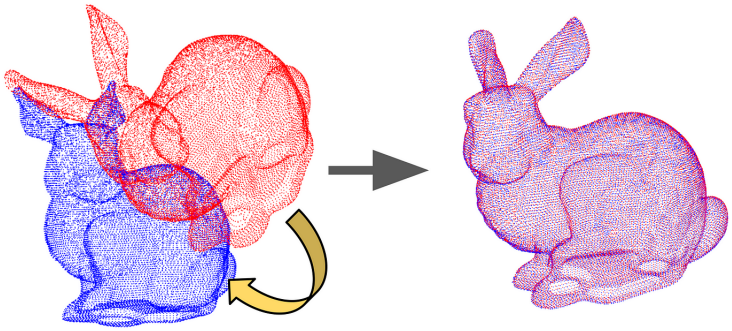


Figure 1.8: *Aligning two point clouds. Source: Biorobotics Lab at Carnegie Mellon University*[5].

The simplest approach is to use directly the raw data as the map. Typically, for a map built using lidar, it consists of an entire point cloud covering the entire are of the operational domain. Then, methods applied for localization are called registration-based methods [Tam et al., 2013] and take all lidar data acquired online to localize in the pre-built map. These methods compute the transformation to apply to align two point clouds. The Iterative Closest Point (ICP) algorithm [Chen et al., 1992] is widely used. To compute the transformation, the errors between the points of both point clouds are minimized. Multiple variants of this algorithm have been proposed such as point-to-line ICP [Censi, 2008] or Generalized ICP [Segal et al., 2009] introducing a probabilistic model in the minimization. An example of point cloud alignments solvable using ICP is visible in Figure 1.8.

As these methods typically involve matching between different point clouds, they tend to be quite storage-intensive. This is why it is necessary to implement other approaches. An approach is to discretize the entire environment into occupancy maps, defined by grid cells or voxels [Oh et al., 2016; Hornung et al., 2013]. These approaches, while less storage-intensive, remain costly for large environments because the entire environment needs to be discretized, requiring sufficient resolution for accurate localization.

Approaches have been proposed to optimize the use of occupancy grids. Typically, [Li et al., 2016a] proposed an offline map builder using lidar scans to provide a map of planes perpendicular to the roads similar to the one stored in RoadDNA Tomtom map visible in Figure 1.6. Planes correspond to projections of point clouds converted into 3D grid maps. After compression, the map size is significantly smaller than the original point clouds and most existing maps, making it suitable for large-scale applications. Then, a Normalized Information Distance matching is used online to find the similarity between online laser scans and the map, this metric being generally used to compute image similarity.

Even though all the techniques presented here help reduce the size of the map, it typically remains extremely large and not suitable for large-scale use, unless a

[5] https://www.ri.cmu.edu/project/point-cloud-registration-pcr/

system regularly downloads map tiles to avoid having a local copy of the complete map, but this obviously poses other issues.

Other types of maps summarizing the information provided by a point cloud can be used, such as Normal Distribution maps (ND) [Biber et al., 2003; Saarinen et al., 2013]. [Biber et al., 2003] first introduced Normal Distribution Transform (NDT) in 2D space. Rather than matching a newly acquired point cloud to the point cloud map, it matches points to a set of normal distributions characterizing the entire environment. [Magnusson et al., 2007] extended it to the 3D space and proposed variants [Magnusson et al., 2009].

To improve the use of a dense map and avoid directly using raw point clouds for matching, other approaches, notably based on deep learning, are employed. For instance, [Schlichting et al., 2018] proposed learning an auto-encoder from a reference point cloud map and thus creating a feature map of point clouds from the learned encoder to find real-time similarity between the acquired point cloud features and the map features.

All the techniques mentioned above, while they may offer promising results in suitable environments, have disadvantages, ranging from the storage size required for some, to their lack of adaptability to new environments without a complete remapping phase, to their dependence on lidar sensors alone, or even on the specific lidar model used, as in the method proposed by [Schlichting et al., 2018], where the learning is valid only for the specific lidar sensor used. Moreover, these techniques mainly work in relatively closed or urban environments and may be significantly limited in open areas or environments with highly repetitive patterns.

Similar approaches with dense maps for cameras exist. This is notably the case with the method proposed by [Herb et al., 2021] using a dense semantic map built from segmented point clouds transformed into meshes to align real-time acquired images with it. [Stumberg et al., 2020] proposed a method of relocalization based on dense visual descriptors obtained from a trained auto encoder used to align images in real-time with previously acquired images. These approaches, just like those used for lidar, are also limited and sensitive to the sensor, or even to the setup configuration.

1.2.3 *Simultaneous Localization and Mapping*

As previously mentioned, employing dense maps as a prior for localization comes with inherent limitations, including the storage requirements, the necessity of an acquisition process and of the choice of a suitable map. Compatibility with the chosen sensor is also crucial; an overly dense map compared to the sensor's resolution can pose challenges.

These maps primarily manipulate raw sensor data or closely related features. In this case, why rely on such maps? Why not leverage the available sensor data directly in real-time, eliminating the need to choose one map over another for

accurate localization? Furthermore, dense maps serve only for localization and lack utility for other functions, such as navigation.

This is the main goal of the Simultaneous Localization and Mapping (SLAM) problem [Smith et al., 1988], which aims to simultaneously build a map of their surroundings and localize themselves within the map in real-time, all without relying on pre-existing maps. This enables localization relative to the observed environment while retaining data about it.

Using raw data allows estimating vehicle's odometry between two frames of acquisition and building a map using raw data help correct the pose regularly to mitigate odometry errors as done by the Lidar Odometry And Mapping (LOAM) approach [Zhang et al., 2014].

Since it uses raw data from sensor, same registration methods than previously introduced with dense maps can be applied as the ICP [Mendes et al., 2016] to build a map by matching newly acquired data with the stored one.

Similar approaches can be applied using cameras as proposed using ORB-SLAM for monocameras [Mur-Artal et al., 2015] where key visual features are extracted from images to build the maps and matching with key images is done. [Mur-Artal et al., 2017] proposed an extension for stereo and RGB-D cameras allowing 3D reconstruction and the generation of a map of features usable for localization after map completion.

[Jensfelt et al., 2006; Lemaire et al., 2007] also proposed vision-based SLAM methods with 3D map of features extracted from images, but managing the case of monocular cameras providing only partial observations of features in the 3D world: bearings. Manipulating sensors providing only bearings between features and the moving vehicle is a complex task, adding partially observed features in the mapping part [Bekris et al., 2006; Huang et al., 2007; Lategahn et al., 2014]. The lack of depth information introduces a scaling problem when constructing a map based solely on monocular vision.

The main difficulty when applying SLAM is error accumulation. In fact at each odometry estimation, a substantial error can be committed, leading to drifts after multiple frames. The map data can end up being distorted or shifted, making subsequent matching more challenging. If the vehicle follows a square trajectory, as the error accumulates, the starting and ending points of the vehicle no longer match. This is the so called loop-closure problem. An example of loop-closure is visible in Figure 1.9.

These approaches also encounter the same problems as dense maps. They are tailored to a single sensor, the computation time can vary, matching errors can occur, and the map size can start to grow. Furthermore, in reality, these SLAM approaches can generally be used to generate dense maps that are subsequently used for localization. However, since they are produced offline they are generally more suitable.

Some researchers proposed multimodal SLAM methods to avoid relying solely on a single sensor and to enhance the robustness of their localization approach [Chen et al., n.d.; Shao et al., 2019; Chghaf et al., 2022; Lin et al., 2021].

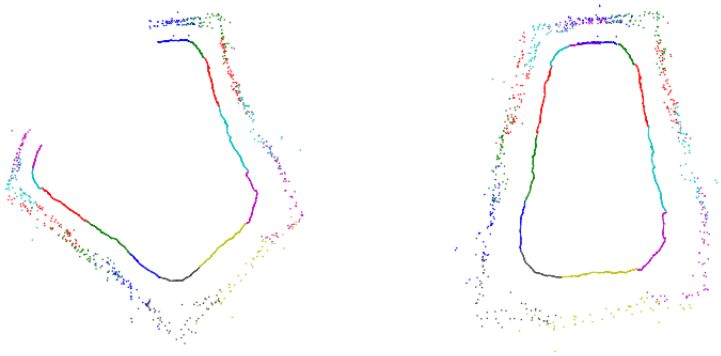


Figure 1.9: *Loop-closure problem from [Williams et al., 2008]. On left, original map. On right, corrected map.*

As done with dense maps, researchers tried to optimize their maps by using feature representation of point clouds [Dubé et al., 2017]. Even, [Hrustic et al., 2020] suggested avoiding reliance on low-level features extracted from data that lack informativeness and robustness under diverse conditions, and instead using landmarks, such as traffic signs, for SLAM.

Consequently, SLAM is commonly used in scenarios where pre-built maps are unavailable or insufficient, such as in exploration robotics or in environments with dynamic obstacles. SLAM algorithms typically rely on onboard sensors like lidars, or cameras to continuously update the map and estimate the vehicle's position relative to it. While SLAM can provide localization without the need for pre-built maps, its accuracy may vary depending on the quality of sensor data and environmental conditions.

That is why, HD maps are generally preferred in autonomous vehicles for their accuracy, reliability and their ability to provide sufficient information for navigation. However, as previously mentioned, dense maps, corresponding to the base map defined by [Elghazaly et al., 2023] are generally insufficient, not well adapted for all vehicle architectures. That is why, HD maps with a higher level of abstraction are necessary: these are vector maps.

However, SLAM can be used to build the point cloud/feature layer of the HD map and can help in environment where the stored map is outdated before new updates.

1.3 HIGH-DEFINITION VECTOR MAPS

By taking the highest level of abstraction possible, an HD map can simply be a set of precisely localized points and segments, thus constituting a vector map [Bao et al., 2022; Poggenhans et al., 2018]. These maps can represent various road features or markings and lane borders with their semantics to simplify their identification. The previously introduced TomTom and HERE HD maps belong partially or completely to this category.

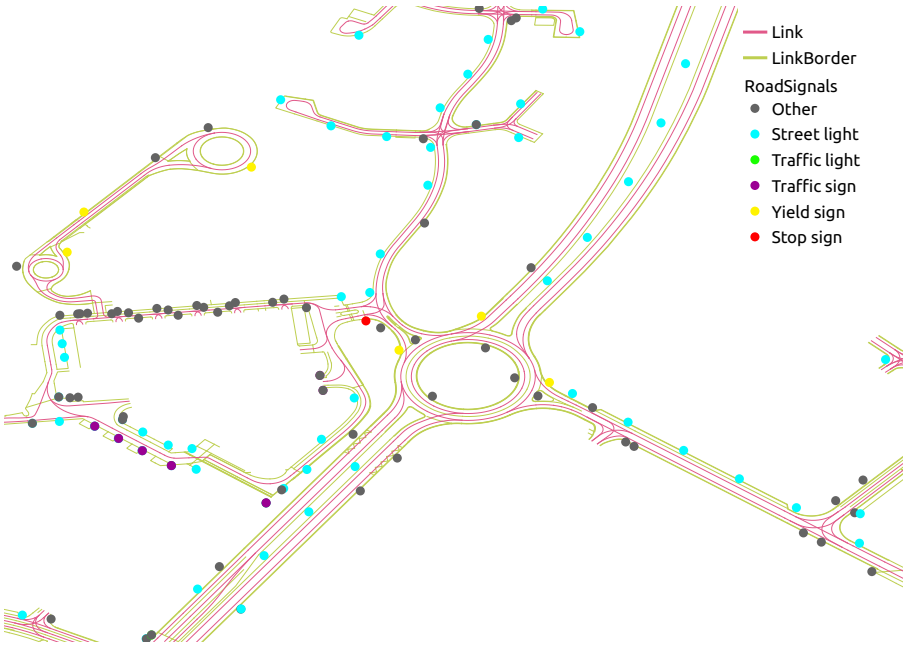


Figure 1.10: *Example of a high-definition vector map*

The advantage of vector maps for localization lies in their consistent representation across different formats, all leading to the manipulation of vector data. Consequently, switching between different maps poses minimal challenges compared to the complexities associated with dense maps that necessitate significant revision of the perception approach when switching maps. In reality, while the conversion of topological layers can pose challenges during format transitions, it is not applicable within a localization context. Their sensor-agnostic nature makes them really suitable for localization with any kind of sensors [Chalvatzaras et al., 2023].

The HD vector map available at the laboratory is visible in Figure 1.10. It is a 2D map containing an ensemble of segments called links defining the topological layer of the map for the navigation. Links contain metadata to help navigation, as speed limit and recommended speed. Link borders are segments used to define road markings, sidewalks, and road edges. Metadata is provided for each link border to determine its type. This information can be useful for localization, especially for lane-level positioning, if the vehicle is capable of detecting these elements.

Finally, the map also contains all the road signals in the road environment. They are characterized by their positions and types. Thus, traffic lights, traffic signs (specifically yield and stop signs, which have their own categories), streetlights, and other road elements are mapped. This map, primarily designed for autonomous navigation purposes, classifies as traffic signs all essential signs for autonomous vehicle navigation, such as no-entry signs. Therefore, all other signs, including directional signs, city entrances and exits, or any other signs with purely informational purposes, are categorized as 'Other'.

The position of a road signal is encoded by a 2D point corresponding to the center of the sign projected onto the 2D plane for road signals consisting of a sign,

or by a 2D point corresponding to the center, in the direction of travel, of the base of the pole.

The acquisition of this map was carried out by professional cartographers at the end of the year 2020 using highly dense lidars and a high-accuracy localization system based on RTK. The various elements of the map were automatically extracted from the acquired data, and any necessary corrections were made. Due to the use of automated methods for extraction and corrections or additions by humans, there may be some errors in the data. This is particularly the case with the road signals category and especially the 'Traffic sign' and 'Other' classes, which are not always correctly defined.

However, a characteristic of almost all road signals stored in the map is the presence of a pole to hold the signals. Furthermore, bollards were explicitly excluded once they were correctly identified. This choice is justified by the fact that they are extremely fragile structures more prone to damage than others, and therefore often removed or moved. Additionally, new ones are regularly added. Thus, if they were retained, this part of the map would be quickly subject to numerous errors.

The other road signals change relatively rarely in comparison, although errors in the map can also appear quickly, especially missing elements on the map (new signs) or as a result of major road works. The other road features are also prone to numerous errors in the map. Indeed, lane markings are certainly, after bollards, the features most prone to map errors. Over time, they tend to disappear quickly, rendering the map deprecated. When they are repainted, they are not necessarily repainted in the same location. Thus, detecting markings may lead to inconsistent information with that stored in the map. Additionally, as with road signals, new markings may be added, or major road works may be carried out

Hence, an HD map becomes deprecated after few years if it is not regularly updated, and maps like those created by the laboratory are complex to update without conducting a new survey by cartographers, which proves to be extremely costly.

The available map contains several errors in various sections due to numerous works, affecting all elements stored in the map as visible in Figure 1.11. However, based on observations, the category of road features least affected on the entire map after multiple years seems to be road signals. As introduced earlier, manufacturers are now heavily focused on HD vector maps updates, and more and more efforts are being made in this direction to counterbalance this issue.

There are multiple ways to exploit HD maps to enhance localization. Some methods are straightforward and do not involve exploiting the connection between perceived elements and stored landmarks. They simply refine the estimated vehicle pose by projecting it onto the map and adjusting it, assuming the vehicle is most probably on a road [Bauer et al., 2016; Li et al., 2017]. However, the potential of correction of these techniques is limited and they suppose a sufficiently accurate initial estimate. Maps can also help to know where to find features in the environments and enable focus perception in the interesting areas to finally

Figure 1.11: *Differences in the environment: At acquisition time (left) vs. present day (right) (Extracted from Google Street View)*

help localization and vehicle guidance [Tessier et al., 2010; Aynaud et al., 2017]. These approaches also assume an initial position sufficiently accurate, but also the absence of significant map errors in the selected areas of interest. Moreover, the interest may prove rather limited depending on the perception approaches implemented, where nowadays, it is easy to build road furniture or lane markings detectors using cameras or lidars and exploiting all received data, while ensuring execution time that meets real-time constraints.

This brings us to the landmark-based localization problem of detecting road elements and associating them with landmarks stored in a vector map, thereby significantly help localization.

1.4 LANDMARK-BASED LOCALIZATION: PROBLEM STATEMENT

We consider here that only passive landmarks are georeferenced in the HD map and not actives beacons like LTE stations that can provide ranging for instance. HD vector maps provide localization data and enable vehicle pose estimation. However, estimating a pose from both map elements and detected landmarks poses many challenges that are briefly presented in the following

1.4.1 *Landmark perception for localization*

To exploit the full potential of vector maps, it is essential to develop road features detectors capable of identifying various landmarks in the environment by using available raw data effectively. As previously mentioned, we focus on monocular cameras and lidars which play a crucial role in autonomous vehicle architectures.

These sensors have their own capabilities that can be exploited differently to provide complementary information. Lidars by emitting laser pulses and measuring the time between emission and reception after hitting objects provide accurate distance measurements and a detailed 3D representation of the world through

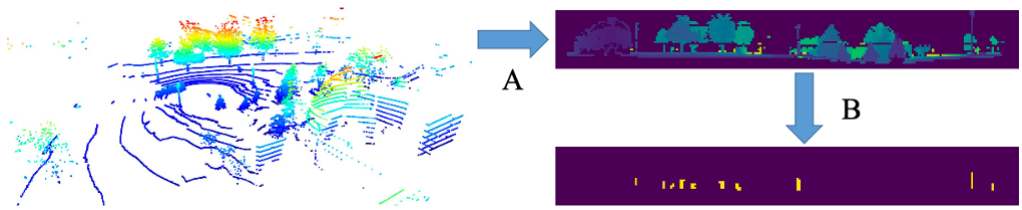


Figure 1.12: *Range image generation (A) from point cloud and poles extraction (B) from [Dong et al., 2021]*

point clouds. Each point provides 3D and intensity information. Lidars typically employ multiple lasers on a rotating platform to scan the environment, offering high accuracy in 3D perception. However, the resulting point clouds may be sparse and limited in range, and lidars can be sensitive to disturbances due to rain, fog, and snow.

Monocular cameras capture 2D images composed of pixels providing information on observed colors and light intensity at specific points. While the information provided is denser and richer compared to lidar, it is constrained by the camera's field of view. However, they are cost-effective and widely available but may struggle in low-light conditions and lack the accurate depth measurement capabilities of lidars.

Given their own characteristics, adapted detection approaches for various types of objects were developed. In the first stage, researchers aimed to exploit the raw data from the sensors to extract interpretable information and analyze the geometric properties of the observed structures.

For lidars, extracting the most intensive elements from the environment applying geometric methods is relatively straightforward. For example, since traffic signs are highly reflective surfaces, intensity data can be used to extract them through intensity thresholding and by applying geometric constraints to the remaining points [Ghallabi et al., 2019]. For example, [Riveiro et al., 2016] proposed an approach based on optimal intensity threshold estimation, clustering using DBSCAN algorithm [Ester et al., 1996] and clusters refinement.

Multiple methods exploit the 3D organization of the point cloud to extract generally pole-like features, using their specific shape characteristics : vertical, thin, isolated, generally high-elevated objects located near the road borders [Riveiro et al., 2016; Li et al., 2021; Dong et al., 2021; Schaefer et al., 2019; Lehtomäki et al., 2010; Rodríguez-Cuenca et al., 2015; Sefati et al., 2017]. For example, to exploit their isolation property, [Li et al., 2021] detected pole-like features by rasterizing the environment and extracting the cells where the point density is higher than their neighbors and checking the connections between the candidates cells. [Dong et al., 2021] transformed the point cloud into range images and exploited the depth difference between poles and their background. An example of range image to extract poles is visible in Figure 1.12.

Applying geometric constraints, it is also possible to extract lane markings [Ghallabi et al., 2018] or road curbs [Wang et al., 2017; Wei et al., 2020]

using lidar data. Generally, due to data sparsity, detecting these features using only one frame provides poor results and multi-frame detections are generally combined to derive a better detection.

Particularly, [Ghallabi et al., 2018] transformed road point clouds into 2D intensity images, since lane markings are highly reflective, to apply a line detection algorithm for images using Hough Transform [Duda et al., 1972].

In fact, even if it may seem at first glance that geometric approaches are less feasible with camera data, in reality the first computer vision methods developed rely on analyzing the geometric properties of visual data to detect objects and structures in the scene. Multiple geometric characteristics can be extracted as lines with Hough transform, edges [Katiyar et al., 2014] applying techniques like the Canny edge detector [Canny, 1986] or corners using methods as Harris corner detection [Harris et al., 1988].

Lane markings are typically recognizable by their line shape and colors. Consequently, approaches based on Hough transform [Li et al., 2016b; Mammeri et al., 2014] or using their characteristics of bright lines on a darker ground [Lu et al., 2014; Revilloud et al., 2013] have been developed.

For traffic signs detection, color-based or shape-based methods can also be applied [Wali et al., 2019]. In fact, one of the major characteristics of traffic signs are their color, contrasting with the rest of the environment. Since this information is accessible, it can be used to detect them [Lopez et al., 2007; Xu et al., 2019]. However, this has the drawback of being very sensitive to brightness and distance.

Shape-based methods can rely on previously introduced algorithms for forms extraction. [Gonzalez et al., 2011] proposed a method relying on Hough transform for shapes extraction similar to shapes observed for traffic signs. [Sathish et al., 2016] proposed a combined approach where colors are used to extract traffic signs and shapes are approximated to refine the detection.

Besides, features in images can be extracted to help detect objects instead of applying directly algorithms to specifically extract some forms. Histogram of Oriented Gradients (HOG) is a feature descriptor that focuses on the shape of an object by computing occurrences of gradient orientation in portions of an image. Scale-Invariant Feature Transform (SIFT) are used to extract key points in images. These key points remain consistent despite changes in lighting, affine transformations, or noise levels, and include corner points, edge points, bright points in shadowed regions, and dark points in illuminated areas. Leveraging such descriptors as image characteristics enhances object detection capabilities.

Hand-crafted features can be computed from images and used for object detection and recognition by diverse machine learning techniques [Pettersson et al., 2008; Zaklouta et al., 2012; Bahlmann et al., 2005; Sathish et al., 2016].

However, all the methods mentioned have limitations. For lidar sensors, the geometric methods introduced depend on many parameters, empirically set, that depend on the sensor used. Besides, they may encounter challenges in dealing with the variability of road features such as appearance diversity, occlusions by

other objects. Typically for pole-like features detection in general, using only 3D representation and applying geometric constraints also lead to tree trunk detection. Many objects can also be missed due to a wrong parameter tuning or sensor characteristics. Lidars have their own limitations compared with cameras. It is impossible to extract semantics and classify traffic signs as algorithms for cameras do, and the detection distance is clearly smaller.

For cameras, similar parameter tuning problems can occur. The geometric-based approaches often rely on handcrafted features and mathematical models. Even if they have the advantage of being interpretable, it brings many difficulties to handle the complexity and variability inherent to real driving scenes.

Even though some methods introduced earlier rely on machine learning to extract the best possible representation of the environment and thus improve detection or recognition performance, as seen with road signs, relying on handcrafted features necessarily limits the capacity for representation and abstraction in the observed scenes.

Nowadays, deep learning for computer vision has experienced a remarkable surge in popularity. These models excel in learning complex representations directly from raw data. While these representations lack interpretability, they possess a broad abstraction capability and can better capture the complexity of objects whose appearance can vary greatly. Instead of using handcrafted features that attempt to represent an object appearance, deep learning models allow computers to learn representations from vast amounts of data. This approach moves beyond the limitations of handcrafted features, which are inherently constrained by the representation they could offer. They were primarily developed for cameras, but nowadays, many solutions exist also for lidars.

Some neural networks are capable of segmenting the entire image, or point cloud for lidars, by assigning a class to each pixel, or point for point clouds, thereby extracting all objects in a scene. The classes can be extremely varied, and datasets used to train such models, like BDD100K [Yu et al., 2020] or Cityscapes [Cordts et al., 2016], contain dozens of different classes across numerous images. An example of manually labeled image from BDD100K is visible in Figure 1.13.

However, for detecting specific objects in a scene, these methods are generally less effective and slower compared to object detection approaches. These networks, similar to segmentation networks, require a substantial amount of training data, which becomes a significant limitation when public datasets are unavailable, or unadapted to the setup used. In this thesis, we propose to focus on the development of object detectors using machine learning techniques specifically for detecting poles.

1.4.2 *The problem of associating sensor detections with landmarks*

Detecting mapped elements enables vehicle localization by leveraging the positional information provided in the map. Since the map contains the accurate coordinates of road features, identifying the relative position of these features with

Figure 1.13: *Example of manually labelled image from BDD100K [Yu et al., 2020] for image segmentation training. Multiple classes are visible with different colors as pedestrians in red, poles in gray or cars in blue*

respect to the vehicle thanks to perception allows us to deduce the vehicle's localization.

If the detected elements are unique and easily distinguishable, as demonstrated in the method proposed by [Wassaf et al., 2021], the vehicle's localization can be estimated easily, as there will be no difficulty in matching the detected elements to their corresponding map features.

However, landmarks are not distinguishable in a vector map, even with their semantics. Besides when focusing on pole detection, there is no semantics available. Hence, it is crucial to establish the association between perception data and map elements as illustrated by the Figure 1.14.

From the vehicle's point of view, multiple objects are detected, as shown by the blue and purple dots, but only some of them correspond to mapped elements, illustrated by the red dots. Other mapped landmarks, not detected or associated are illustrated with black dots. In addition to the challenge of establishing associations, especially when there are uncertainties regarding the positions of the detections, some of the objects detected can be wrong detections or unmapped objects and should not be associated. Another issue not depicted in this figure is the occlusion of some landmarks by other objects or road users, rendering them undetectable by sensors. While this mainly reduces the number of detections, it can also affect the association process.

Additionally, depending on the detected objects and the road environment, another challenge that may arise is the risk of association ambiguities as illustrated by the Figure 1.15. When detecting pole-like features, various situations can arise. For example, as shown in Figure 1.15a, when two poles are detected along a single-lane road with poles regularly spaced on each side, it creates a scenario with multiple possible vehicle poses due to the ambiguity in the scene, potentially leading

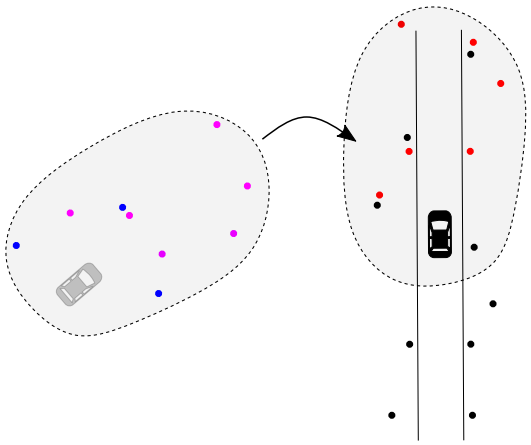


Figure 1.14: *Data association problem. On the left, the vehicle's surroundings are shown with detected elements in blue and purple. On the right, the actual vehicle position is displayed, with undetected map elements in black and detected map elements in red. The purple dots are the detections corresponding to mapped elements in red and must be associated with them. Other detections should be discarded.*

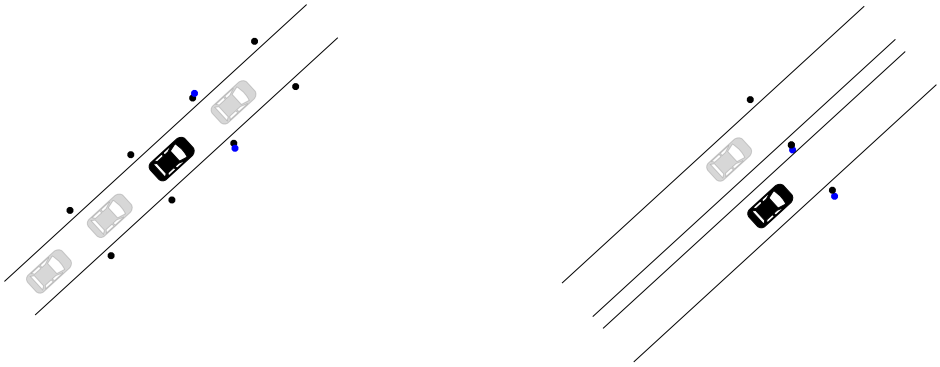
to errors along the road. However, in this context, the poles can still help reduce lateral positioning errors. Figure 1.15b depicts a multi-lane scenario where detecting two poles can result in lateral ambiguity, making it challenging to accurately determine the correct lane.

As a result, there is a risk of incorrectly associating detections with map features, leading to inevitable positioning errors. Some researchers attempted to manage the risk of incorrect associations, as noted in [Joerger et al., 2016; Arana et al., 2020]. However, the probability of incorrect associations increases rapidly in cluttered environments. This challenge is primarily due to the lack of distinguishability among detected objects.

To address this, [Joerger et al., 2017; Hassani et al., 2023] proposed improving the distinguishability of detected objects and prioritizing the most distinguishable ones for association. However, this approach requires modifications to the map and is limited to the lidar used making it unsuitable for sensor-agnostic methods.

Additionally, prioritizing only the most distinguishable features can limit localization system availability by losing too many associations. [Nagai et al., 2024] showed that maintaining a sufficient density of landmarks is crucial.

In general, avoiding incorrect associations is impossible without perfectly distinguishable landmarks. While error-free data association is crucial for map correction tasks, for localization performance, it may be sufficient to ensure that data association contributes meaningfully by minimizing potential misassociations and maintaining a sufficient density of detected landmarks. It is effective as long as it does not degrade localization performance or if the system can detect some localization errors online. The accuracy of detections corresponding to mapped landmarks improves the refinement of localization and avoidance of erroneous associations.



(a) *Along-track ambiguities due to pole-like landmarks.*

(b) *Cross-track ambiguities due to pole-like landmarks in a multi-lane context*

Figure 1.15: *Examples of potential positioning ambiguities due to association between map information (in black) and detections (in blue). The true vehicle pose is visible in black and potential other poses are visible in gray.*

Therefore, the goal is to maximize the detection of map elements while minimizing incorrect detections. To achieve this, a perception pipeline tailored to each sensor and specifically designed for landmark-based localization, trained to recognize only map features, is essential. Although this approach may occasionally identify unmapped elements resembling those on the map, such cases are inevitable but could potentially be minimized.

1.4.3 *Localization with landmarks*

Consider a simple scenario where we know which map elements have been detected. Consequently, we know their accurate positions from the map, and their positions relative to the vehicle. This allows for the estimation of a pose. For example, [Betke et al., 1997] employed a monocular camera and discernible indoor features stored in a map to estimate pose using a least-squares approach. However, these approaches are feasible only if a sufficient amount of detections is available (three in this example).

In a more realistic scenario, this approach may face significant limitations. Estimating the vehicle's pose might not always be feasible, as detectors can miss detections or occasionally produce excessive errors or uncertainties which pose challenges for snapshot estimations. Generally, vehicle pose estimation involves fusing data from multiple sensors using a filter, which delivers frequent updates on the vehicle's pose and accounts for its dynamics.

With this method, even a single detection can potentially enhance the vehicle's pose estimation. This is illustrated by the example in Figure 1.16. An initial pose is expressed in a global frame O, relative to a given body frame B, and a map feature is visible, with its coordinates expressed in the same global frame O. This feature is detected by a lidar, as illustrated by the blue dot, whose coordinates are

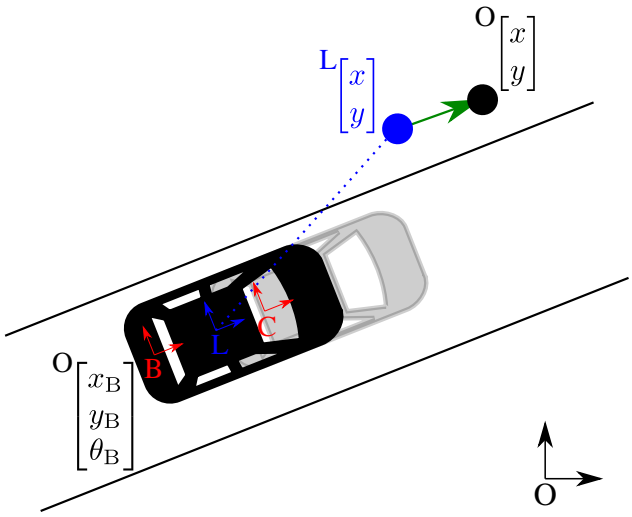


Figure 1.16: *Landmark-based localization. Map element coordinates are expressed in a global frame O. The coordinates of a detected element from a lidar sensor are expressed in the L frame and highlighted in blue. Additional frames, such as the C frame for another sensor and the vehicle's body frame B, are shown in red. The vehicle's prior pose, indicated by the black car, is corrected by associating the detection with the map element. This correction, represented by the green vector, updates the pose to the new position, shown by the grey car.*

expressed in the lidar frame L. By associating the detection with the map feature, a correction can be estimated and applied to the pose estimate.

In a multi-sensor system, each sensor collects different measurements, which means these measurements can be expressed in different frames. Therefore, a common reference frame, typically aligned with the vehicle's pose (denoted as B here), is used. Each sensor has its own reference frame, such as L for the lidar and C for a camera, for example. To accurately apply corrections, it is essential that all sensors are precisely calibrated, with exact transformations from each sensor's frame to the vehicle's frame B.

1.5 CONCLUSION

To improve localization performance, the inclusion of maps and perception data is essential. Due to limitations of digital maps, a new kind of map has been introduced: the HD Map. HD maps are extremely accurate representations of the environment and can be of multiple types, containing dense information or pointwise elements and segments. Due to their sensor-agnostic capabilities, lightweight design, potential for generality, and ability to provide accurate information, HD vector maps are very interesting to enhance localization capabilities of autonomous vehicles. To be used, object detection approaches for various sensors compliant with objects stored in the map are necessary.

The sensors must be varied for the system to operate effectively under different environmental conditions and provide as much usable information as possible

alongside the map. Detectors must be developed to detect as many elements as possible stored in the map. For this purpose, lidars and cameras are particularly suitable and complement each other, offering a wealth of information and versatility.

While traditional approaches exist for object detection that do not rely on extensive data training, they often yield limited performance, require meticulous difficult tuning, and are highly sensitive to the sensor type and environmental conditions. These days, machine learning and particularly deep learning have become increasingly widespread, enabling the development of more effective object detectors. However, a large amount of data is now required. Depending on the detection task, public datasets may not be readily available, such as in the case of poles for camera-based detection. Additionally, the learned models are data-dependent and may be less adaptable to different sensors, especially in the case of lidar, or to unseen scenarios.

To fully leverage the map, it is essential to develop detectors capable of robustly identifying a wide range of landmarks and ensuring their effortless adaptability. For a given autonomous vehicle setup, it is crucial to adapt to new conditions encountered, particularly in unfamiliar road environments, which often come with new maps. In this context, machine learning emerges as the most suitable approach. By providing data that covers the road environments represented in the map, models for detecting mapped landmarks can be trained. When adapting to a new environment, additional training on newly acquired data from that environment may be required. Then, beyond being designed to detect map elements, perception models must be developed to be as adaptable as possible without adding additional human effort. Typically, machine learning models require annotated datasets, which can be costly. Adapting a model to new road conditions often necessitates new labelled data, increasing overall costs. Therefore, minimizing or avoiding human annotation is essential.

Here, perception serves to enhance localization through data association with the map. Detection is therefore focused on identifying map elements, aiming to minimize data association errors by detecting as many map elements as possible while avoiding those absent from the map.

Chapter 2

Map-driven Automatic Annotation for Pole-like Feature Detection

Contents

2.1 Introduction

The importance of detecting landmarks stored in HD vector maps for accurate localization for AD has been well-established. Commercial products such as MobilEye cameras provide detection of some landmarks such as traffic signs or road markings. They rely on the use of deep neural networks trained on a substantial amount of data that have been manually annotated. In this thesis, we aim at building a detector for poles which are fundamental elements of the road infrastructure and are well distributed in the environment. We wish to achieve such a goal with

the use of state-of-the-art machine learning based techniques but without the burden of manually annotating a large amount of data.

In this chapter, we propose an automatic annotation approach leveraging the prior knowledge provided by an HD vector map. We first show how to project map data onto images and how to correct and refine this step with the help of a lidar sensor. To account for the limitations due to the use of maps, we then show how to improve the annotation thanks to a multi-modal approach with pre-trained image and lidar semantic segmentation neural networks. The results are experimentally validated on some data acquired in Compiègne and manually annotated.

2.2 AUTOMATIC ANNOTATION FOR MACHINE LEARNING: STATE-OF-THE-ART

Training deep neural networks requires a significant amount of annotated data. For that, depending on the object detection task, multiple datasets are available. Since training of a deep learning model is particularly long, they can be pretrained on extremely large datasets containing numerous images of various objects as the MS COCO dataset [Lin et al., 2015] containing more than 200 000 images. Afterward, an initial usable initialization of the network weights is obtained, and the model can be fine-tuned to be used for the desired task.

With existing datasets, it is possible to build detectors useful for vehicle navigation, and potentially for localization as well. For example, numerous datasets for detecting traffic signs are available containing traffic signs from around the world [Ertler et al., 2020] or from specific countries [Houben et al., 2013; Timofte et al., 2014]. However, for some types of objects, no datasets are available for object detection tasks. For example, from our knowledge, there is no widely available dataset for pole detection using cameras mounted on a vehicle.

Generally, new datasets are consequently created through manual labelling which becomes particularly costly when the amount of data increases. When no annotated data is available, it is consequently interesting to provide an automatic annotation method. Researchers proposed automatic and semi-automatic annotation methods to build datasets [Dong et al., 2023; Sun et al., 2020; Lee et al., 2021; Yu et al., 2016; Qi et al., 2021] providing hard labels, to be used in classical supervised training pipelines. Hard labeling means that there is no quantification of the uncertainty of each label. In particular, pseudo-labeling, a process that involves annotating unlabeled data by using predictions obtained from a pre-trained network for subsequent retraining, is generally applied [Lee, 2013; Sohn et al., 2020; Yang et al., 2021; Radosavovic et al., 2018]. However, the pre-trained network must be consistent with the detection task at hand and pseudo-labeling can be insufficient or lead to poor results due to the low quality of labels.

These approaches are prone to errors and can lead to lower performance compared to annotations made by humans if the quality of the automatic annotations is not properly controlled which will decrease the performance of the learned model.

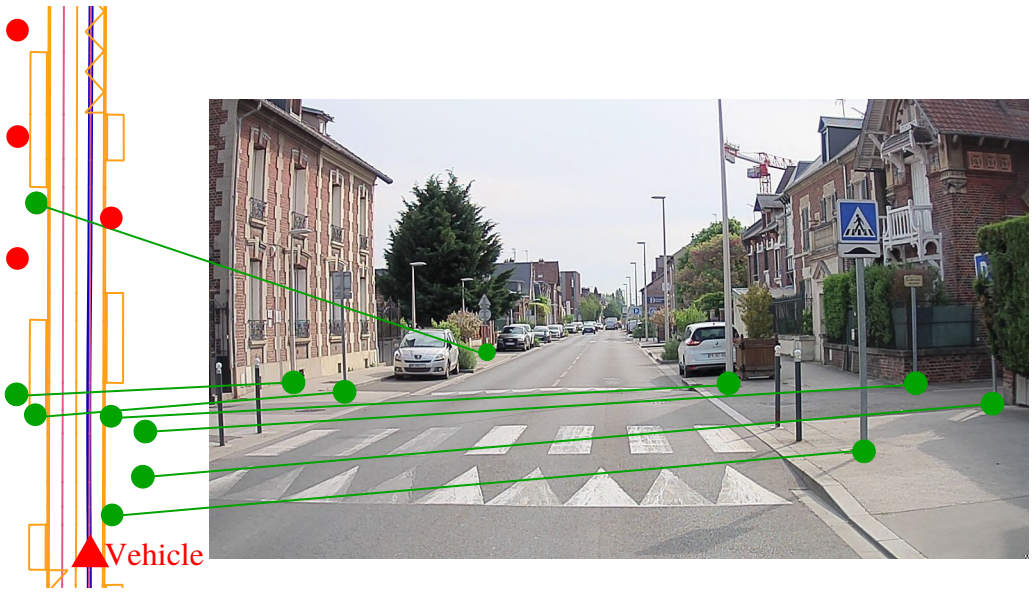


Figure 2.1: *Illustration of the connections between georeferenced poles on the map (left) and an image captured by a vehicle (right). The pole bases in red are not visible from the camera.*

To get good annotations automatically, multi-source approaches can be employed. For instance, in [Tsai et al., 2023], the authors employed a set of pre-trained multi-frame 3D detectors in lidar point clouds and fused their detections to build new pseudo-labels on an unlabeled set.

2.3 MAP-BASED AUTOMATIC IMAGE ANNOTATION

The 2D HD vector map introduced in Section 1.3 provides prior knowledge about the driving scene, in particular the position of the poles in the vehicle's surroundings. In the map, the poles are encoded as a 2D point without height information. As such, they are interpreted as representing the coordinates of the pole's base at ground level. Figure 2.1 illustrates the correspondence between the 2D vector map pole features and the image captured by a vehicle. Different challenges are pictured in this figure:

- Proper projection of the map features onto the image requires accurate localization of the vehicle, a proper extrinsic and intrinsic calibration of the camera with respect to the vehicle body frame and an estimation of the ground plane.
- The bases of some poles are occluded (red dots) and need to be removed.
- Some road features such as the bollards that can be seen on the side of the pedestrian crossing are visually close to poles, but should not be detected, as they are not mapped.

2.3.1 *Projection of 2D HD vector map features onto images*

In our 2D HD vector map introduced in Section 1.3, a pole i is encoded as a 2D point ${}^{O}P_i = \left[{}^{O}x_i, {}^{O}y_i\right]^\top$ expressed in the map frame denoted by the superscript O. The map composed of all the poles is therefore a set of n 2D points ${}^{O}\mathcal{M} = \left\{{}^{O}P_i\right\}_{i=1,\dots,n}$. The map frame O is defined as an East-North-Up (ENU) navigation frame with its origin in the vicinity of the navigation environment. It defines a Cartesian spatial reference system formed from the tangent plane defined by the local vertical direction and the Earth's axis of rotation fixed to a specific location and is location dependent. The transformation from geodetic coordinates to ENU coordinates, in particular when using different localization sources, is explained in Appendix B.

To project the map points onto the camera images, their coordinates need to be transformed into the camera frame C. We do it in two steps by first transforming them into the vehicle body frame B (these frames are pictured in Figure 1.16). Because the map is in 2D, a naive approximation is to assume that the ground is flat and to redefine a pole i as a 3D point lying on the ground ${}^{O}P_i = \left[{}^{O}x_i, {}^{O}y_i, 0\right]^\top$. Additionally, we assume that the height h_B of the vehicle body frame with respect to the ground is known. By using homogeneous coordinates, the coordinates ${}^{B}P_i$ of the map point in the vehicle body frame can be written as

$$\begin{bmatrix} {}^{B}x_i \\ {}^{B}y_i \\ {}^{B}z_i \\ 1 \end{bmatrix} = {}^{B}T_O \begin{bmatrix} {}^{O}x_i \\ {}^{O}y_i \\ 0 \\ 1 \end{bmatrix} \tag{2.1}$$

where

$${}^{B}T_O = {}^{O}T_B^{-1} \quad \text{with } {}^{O}T_B = \begin{bmatrix} \cos\theta_B & -\sin\theta_B & 0 & x_B \\ \sin\theta_B & \cos\theta_B & 0 & y_B \\ 0 & 0 & 1 & h_B \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

The vector $[x_B, y_B, \theta_B]^\top$ corresponds to the vehicle pose in the map ENU frame and ${}^{B}T_O$ corresponds to the rigid transformation from the map frame to the vehicle body frame. Then, similarly, the coordinates of the map point can be transformed into the camera frame as follows:

$$\begin{bmatrix} {}^{C}x_i \\ {}^{C}y_i \\ {}^{C}z_i \\ 1 \end{bmatrix} = {}^{C}T_B \begin{bmatrix} {}^{B}x_i \\ {}^{B}y_i \\ {}^{B}z_i \\ 1 \end{bmatrix} = {}^{C}T_B\,{}^{B}T_O \begin{bmatrix} {}^{O}x_i \\ {}^{O}y_i \\ 0 \\ 1 \end{bmatrix} \tag{2.3}$$

where ${}^{C}T_B$ is the rigid transformation from the vehicle body frame to the camera frame. Finally, the coordinates in the image frame are computed as follows:

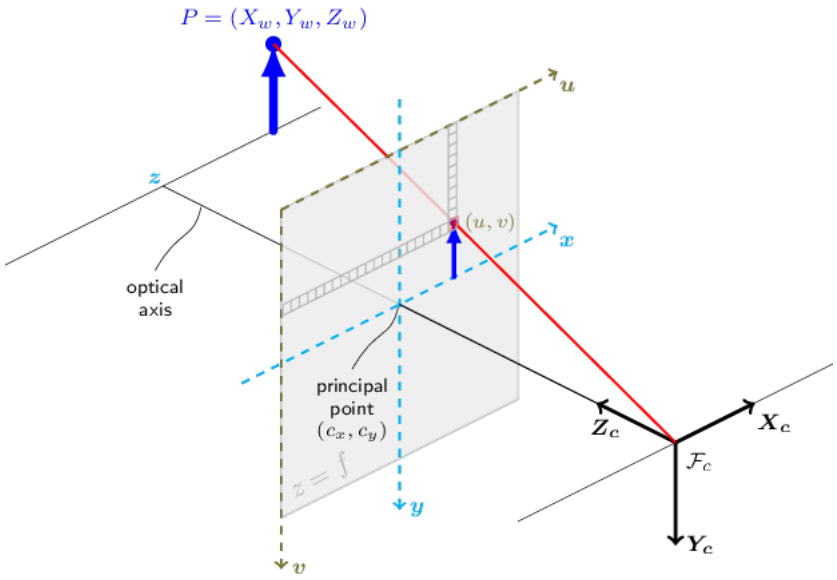


Figure 2.2: *Projection of a 3D point expressed in camera frame onto an image. (Source: OpenCV[1])*

$$ {}^{I}P_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} f_x {}^{C}x_i / {}^{C}z_i + c_x \\ f_y {}^{C}y_i / {}^{C}z_i + c_y \end{bmatrix} \tag{2.4} $$

where the camera intrinsic calibration parameters are the principal point (c_x, c_y) and its focal lengths (f_x, f_y). Note that the camera calibration can be extended to include distortion parameters. The projection in the image frame is pictured in Figure 2.2. Note that, in the camera frame, the z-axis is commonly defined as the depth direction and not the upward one.

To accurately project map points onto an image, it is therefore necessary to have accurate estimates of the vehicle pose, the extrinsic calibration from the camera to the vehicle body frame and the intrinsic calibration of the camera. In the context of automatic annotation, the vehicle pose can be estimated offline with post-processing in order to achieve better accuracy. The proper calibration of sensors such as cameras is an issue in itself, which is outside the scope of this study. In our case the calibration was done manually using external laser-based equipment as well as standard calibration procedures.

By applying all the transformation introduced previously to the whole map, we get the annotations illustrated in the Figure 2.3.

Several challenges arise when simply projecting the map onto the image. Firstly, map poles that fall inside the camera's field of view are annotated, but, given that the map covers a vast area, numerous poles may be too distant from the camera or obscured by buildings, and therefore should not be projected. Then due to the 2D assumption, most of the pole bases are badly projected onto the images. It is due to the fact that the ground generally does not correspond to a plane tangent to the vehicle as visible in Figure 2.4.

[1] https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html

Figure 2.3: *Naive projection of map features onto two images. Orange crosses highlight distant poles that are not visible in the image. Red crosses highlight badly projected annotations since they are not on the ground. Black crosses highlight masked annotations due to occlusion. A missing pole in the map is highlighted with a blue circle in the top image.*

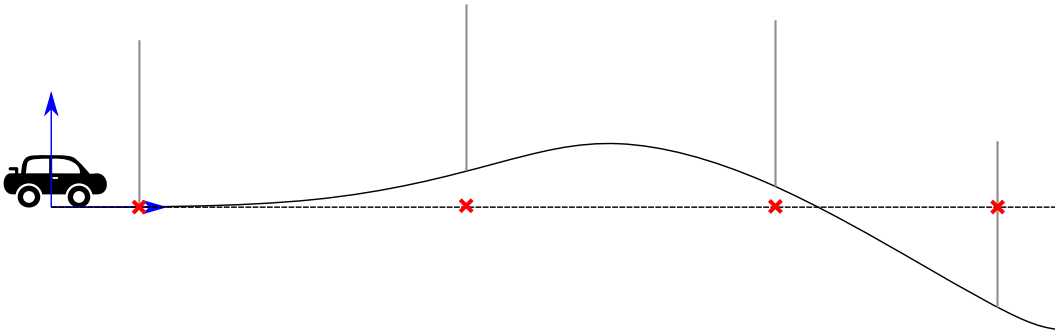


Figure 2.4: *Road profile in 2D from the vehicle point of view with the 2D HD vector map corresponding to red points in the x-axis.*

While the road elevation changes, the projection from the map onto the vehicle frame is always on the tangent plane. Consequently, the 2D assumption of the map works only near to the vehicle or when the elevation variation is significantly low.

Finally, when the poles are close, they can still be occluded by dynamic objects as driving or parked vehicles but also by static elements. Managing these occlusion issues is crucial, as depicted in Figure 2.1, where the desired outcomes are illustrated.

2.3.2 *Ground plane refinement with lidar*

Firstly, to improve the projection, it is mandatory to estimate the ground elevation. For that, as illustrated by the Figure 2.4, we cannot rely only on a unique ground plane, and a precise ground segmentation is consequently needed. We could imagine multiple solutions.

We choose to rely on lidar data for ground segmentation and pole base height correction. Firstly, to manage distant poles we add a distance threshold D_{max} between the vehicle and the poles stored in the map. If the distance is above this threshold, the pole base is not projected onto the image. Similarly to Eq. (2.3), mapped poles are first transformed into the lidar frame L as follows:

$$ {}^{L}\mathcal{M} = \left\{ {}^{L}P_i = {}^{L}T_O \, {}^{O}P_i \middle| {}^{O}P_i \in {}^{O}\mathcal{M} \right\} \tag{2.5} $$

From these points, we extract the nearby poles as follows:

$$ {}^{L}\mathcal{M}_D = \left\{ {}^{L}P_i = \left[{}^{L}x_i, {}^{L}y_i, {}^{L}z_i \right] \in {}^{L}\mathcal{M} \middle| \sqrt{{}^{L}x_i^2 + {}^{L}y_i^2} \leqslant D_{max} \right\} \tag{2.6} $$

Next, the 3D point cloud ${}^{L}\mathcal{P} = \left\{ {}^{L}p_j = \left[x_j, y_j, z_j \right]^\top \right\}_{j=1,\dots,m}$ from the lidar corresponding to a given image is used to have a better estimate of the ground surface by applying the ground segmentation method proposed by [Jiménez et al., 2021]. This algorithm separates all the lidar points into two groups, a ground points one

${}^{L}\mathcal{G}$ and a non-ground one ${}^{L}\overline{\mathcal{G}}$. Due to the impossibility of ground segmentation when points are lacking, we set D_{max} to 50m.

For a given pole base ${}^{L}P_i \in {}^{L}\mathcal{M}_D$, the goal is to find a better estimate of its coordinate ${}^{L}z_i$. First, we extract the lidar ground point near the pole:

$$ {}^{L}\mathcal{G}_i = \left\{ {}^{L}p_j \in {}^{L}\mathcal{G} \mid \left\| {}^{L}p_j - {}^{L}P_i \right\| \leqslant \tau \right\} \tag{2.7} $$

The neighborhood of the pole is defined with a maximum search distance τ around ${}^{L}P_i$. This search distance is defined taking into account that the closer to the sensor, the denser the ground points, and consequently, the search distance must increase when the distance between the sensor and ${}^{L}P_i$ increases. This part is explained in Appendix C. If the neighborhood is empty, ${}^{L}\mathcal{G}_i = \emptyset$, then the pole ${}^{L}P_i$ is removed from the map ${}^{L}\mathcal{M}_D$. From ${}^{L}\mathcal{G}_i$, we define the new pole height as the median height of the ground points:

$$ {}^{L}z_i^* = \text{median} \left\{ z_j \middle| {}^{L}p_j = \left[x_j, y_j, z_j\right]^\top \in {}^{L}\mathcal{G}_i \right\} \tag{2.8} $$

From this corrected height, we construct a corrected map ${}^{L}\mathcal{M}_D^*$ which can be projected onto the image frame, as illustrated in Figure 2.5. The lidar point clouds are also projected onto the images, with ground points highlighted in green and non-ground points in red.

In the top image, only properly projected points are visible, while the occluded one has been removed using the height estimation approach, as no ground point was sufficiently close. However, in the bottom image, despite all annotations being correctly projected onto the ground, some occluded points remain due to the large ground search area. With a refined search area, it may have been possible to reject the occluded points. However, for the bottom image, one of the occluded annotation is close to the ground, making it difficult to remove using this method. Moreover, ground segmentation errors can also occur and some occluded pole bases may be kept. This highlights the need for methods specifically designed to remove occluded annotations.

2.3.3 *Occluded annotation removal*

To remove occluded annotations, we first project all the non-ground lidar points ${}^{L}\overline{\mathcal{G}}$ and the map poles ${}^{L}\mathcal{M}_D^*$ onto the image as pictured in Figure 2.5. By using the projection equation (2.4) to compute the image u-v coordinates, we keep the depth information along the ${}^{C}z$-axis: ${}^{I}\overline{\mathcal{G}} = \left\{ {}^{I}p_j = \left[u_j, v_j, {}^{C}z_j\right]^\top \right\}_{j=1,\dots}$ and ${}^{I}\mathcal{M}_D^* = \left\{ {}^{I}P_i = \left[u_i, v_i, {}^{C}z_i\right]^\top \right\}_{i=1,\dots}$.

Similarly to the height estimation, for each map pole ${}^{I}P_i$, we first extract the non-ground lidar point in its neighborhood: ${}^{I}\overline{\mathcal{G}}_i$. A point ${}^{I}P_i$ is considered occluded if the difference between its depth ${}^{C}z_i$ and the median of depths of surrounding lidar points is above a given threshold $\mathcal{D}_{max} > 0$:

Figure 2.5: *Projection of map pole bases onto two images after height estimation using lidar data. The estimated ground points and non-ground points are highlighted with green and red dots respectively. The correctly projected points are highlighted with green crosses and the occluded pole bases are highlighted with black crosses. A missing pole in the map is indicated with a blue circle in the top image.*

$$ {}^{C}z_{{}^{I}\overline{\mathcal{G}}_i} = \text{median}\left\{ {}^{C}z_j \,\middle|\, {}^{I}p_j = \left[u_j, v_j, {}^{C}z_j\right]^\top \in {}^{I}\overline{\mathcal{G}}_i \right\} \tag{2.9} $$

$$ {}^{C}z_i - {}^{C}z_{{}^{I}\overline{\mathcal{G}}_i} \leqslant \mathcal{D}_{max} \tag{2.10} $$

The idea is that if a map feature is occluded by an obstacle, such as the black cross on the car in the Figure 2.5, the estimated depth from the point cloud will be shorter than the distance obtained from the map data. The threshold controls how strict we are with respect to this difference in distance.

${}^{I}\overline{\mathcal{G}}_i$ is defined by applying a search area around ${}^{I}P_i$ as visible in the top image of the Figure 2.6. Since ${}^{I}P_i$ should correspond to a pole base, it is unnecessary to search for points too far below the annotation. However, given that poles are vertical, thin structures pointing upwards, it is crucial to define a search area that considers these characteristics. Hence, the chosen rectangular shape. This search area is defined by its width w, its height h such that $h = \alpha w$, with $\alpha > 1$ to favor vertical areas, and the vertical offset v of its center relative to the annotation to favor the area above the annotation over the one below.

In Figure 2.6, the search areas are defined by $w = 15\text{px}$, $\alpha = 3$ and $v = 15\text{px}$. As highlighted with black and red boxes, it allows identifying respectively visible and occluded poles leading to the final annotation result visible in the bottom image. When checking for potential occlusions, errors can arise and some visible pole bases can be removed due to the search area. For instance, the red cross in the figure represents a classification error, where a visible pole base was incorrectly identified as occluded. This area could be optimized, potentially reduced when the distance between the pole base and the car increases, however errors will always be possible.

It is consequently important to note that the obtained annotations may still have imperfections due to potential errors in the entire process due to parameter tuning as with the occlusion checking in Figure 2.6, calibration issues, ground segmentation errors, or the presence of unmapped features as in Figure 2.5. Besides, mapped poles could potentially be removed after roadworks producing wrong annotations. Then, using lidar data to check for occlusions implies the lidar and the camera are synchronized with a similar field of view to guarantee they observe the same environment.

2.3.4 *Limitations*

Map-based automatic annotation allows for the creation of annotated images that adhere to the map's pole definition, potentially enabling consistent pole detectors, typically excluding bollards. However, this approach faces numerous limitations.

Firstly, using HD maps can lead to missing annotations if the map is outdated, or it can result in annotations being added where poles have potentially been removed. Secondly, to correctly annotate sensor data, it is necessary to transform the map points into different frames. Therefore, annotation solutions are inherently sensitive to calibration errors or post-processing errors when calculating the poses used for transformation into the vehicle frame. These errors cannot be

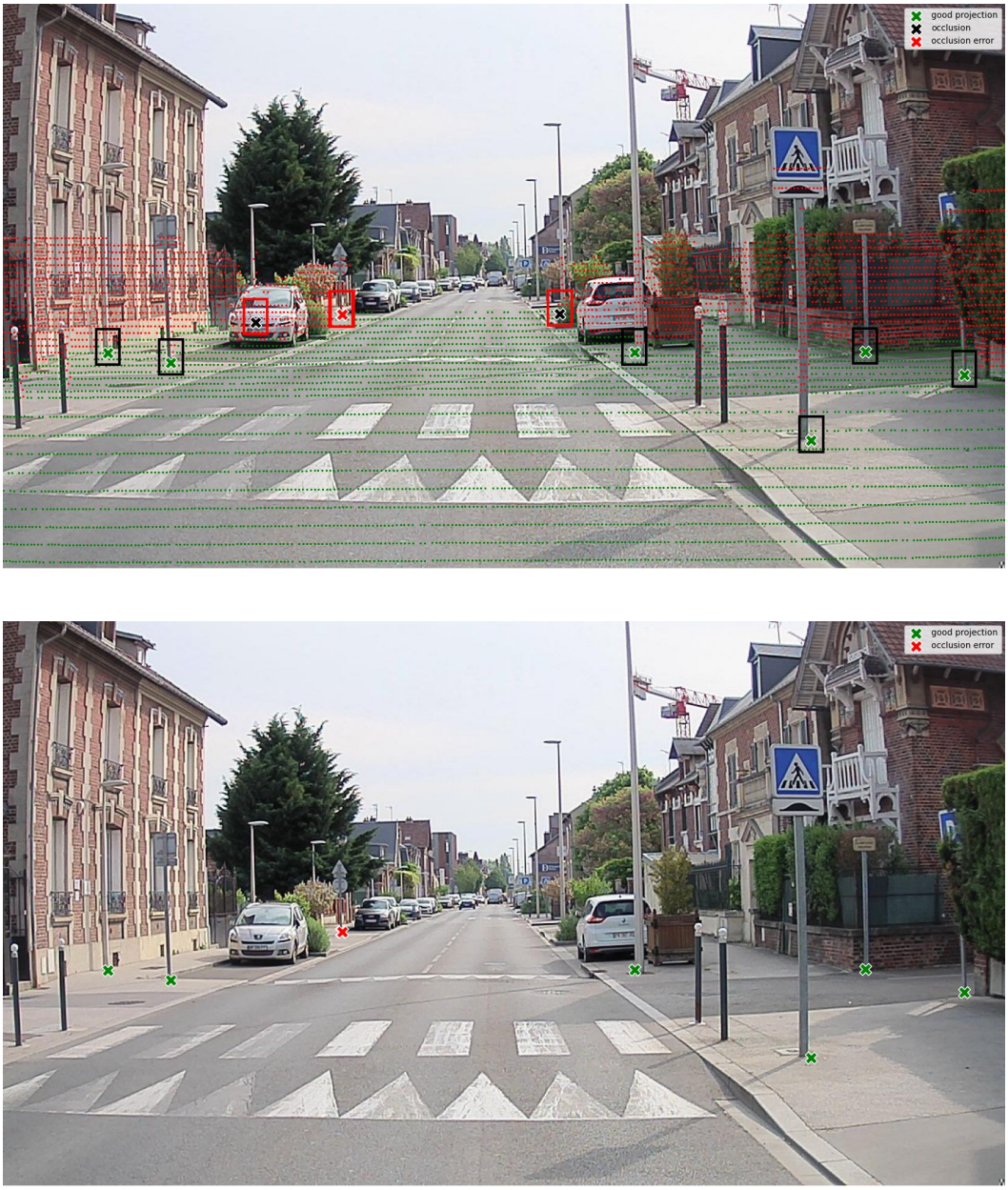


Figure 2.6: *Occlusion checking using lidar point clouds and rectangle search areas. In the top image, the non-ground points and ground points are highlighted with red and green dots respectively. The search areas are indicated with black boxes when identified as visible and with red boxes when identified as occluded. The final result is visible in the bottom image.*

avoided if the sensors are not regularly recalibrated and if the post-processing cannot estimate an accurate pose. As a result, it is highly likely that the obtained annotations will generally be offset in the image relative to the actual position of the poles.

Additionally, geometric methods are implemented to refine the annotations. The final result is sensitive to these methods and their tuning, which is difficult to perfectly adjust for each dataset that might be annotated. The tuning obtained is thus very likely approximate, providing good performance for the annotated elements but not guaranteeing a consistent rate of annotations during processing.

Therefore, relying solely on a map-based annotation method could prove insufficient and necessarily limit detection performance. It could be beneficial to suggest alternative complementary approaches that might be less sensitive to the various challenges outlined. This could help verify the accuracy of existing annotations by correcting any positioning errors in the images, addressing missed annotations from the map-based method, and identifying incorrect annotations caused by errors in the map-based approach. The ultimate goal is to minimize any negative impact on the performance of a detector trained with these annotations.

2.4 MULTI-MODAL POLE ANNOTATION

In this section, we introduce two additional annotation sources using semantic segmentation, one applied on the image and another on the lidar point cloud. Then we study how they can be combined with the map-based annotation.

2.4.1 *Image segmentation-based automatic annotation*

Structures like poles can be seen at the pixel-level from a semantic segmentation point of view. In this setup, all the pixels in an image are assigned a class label. It needs pixel-wise finely annotated images such as the ones proposed in the BDD100K dataset [Yu et al., 2020]. Deep neural networks for semantic segmentation such as the HRNet [Wang et al., 2021] trained on such a dataset are able to detect pole-like structures in images at pixel-level. These semantic segmentation networks are much more computationally demanding than object detection ones. In our case, we use these networks in an offline manner for automatic annotation.

We follow the annotation procedure pictured in Figure 2.7. The segmentation network processes a given image to generate an estimated segmentation mask S that assigns to each pixel an estimated class label.

As we are only interested in detecting the pole bases, we want to keep only the poles that stand on the ground. For this purpose, our approach firstly extracts the estimated ground mask S_{ground} which includes the pixels which label belongs to road, sidewalk or terrain. Subsequently, to get all pole bases relying on the ground we expand S_{ground} by adding N pixels around each pixel from the mask to obtain a new mask $S_{extended}$. This mask includes all pole pixels near the ground, capturing the lowest visible section of each pole. Then, from $S_{extended}$, we ground

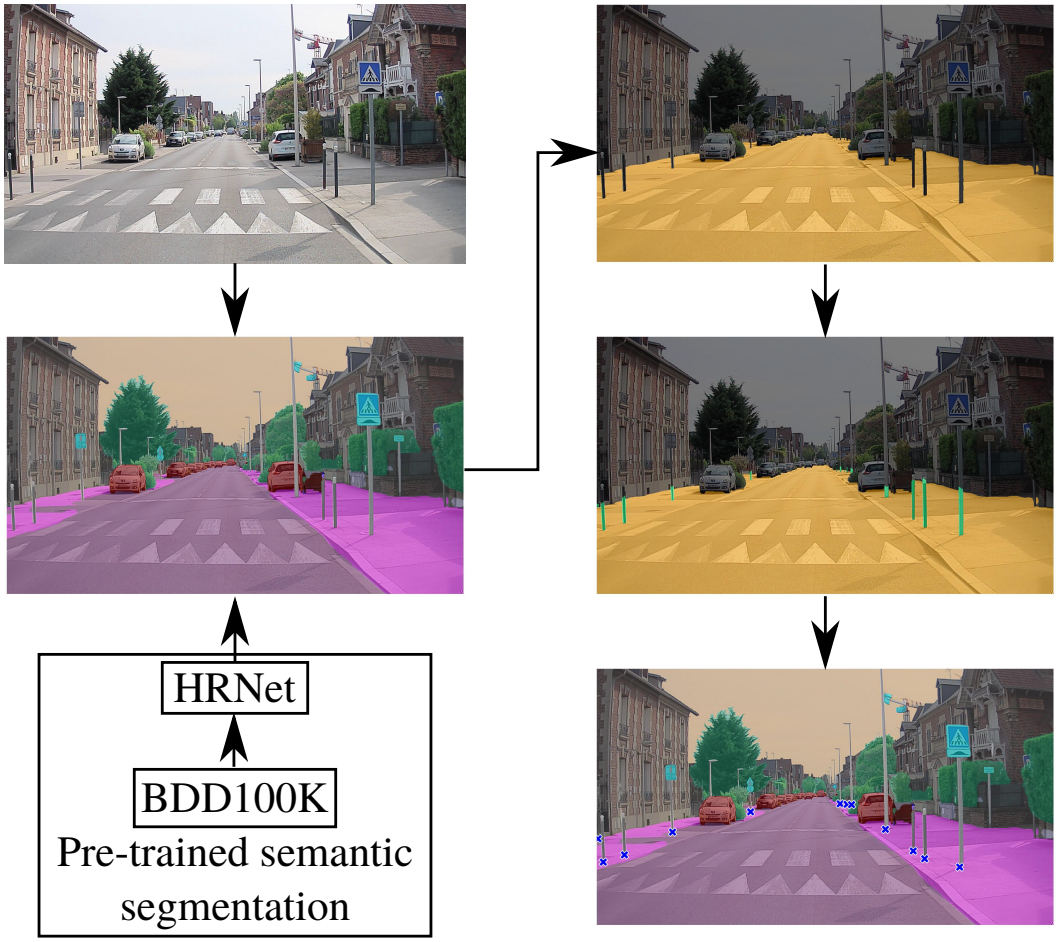


Figure 2.7: *Pole base extraction using semantic segmentation network. The network processes a given image to obtain an estimated segmentation mask. From this segmentation mask, a ground mask is obtained as visible in orange. By expanding the ground mask by few pixels we can extract clusters corresponding to pole bases as visible in green and annotate the lowest parts of these clusters as highlighted with blue crosses.*

into connected clusters all the pixels with pole-related label (in the case of the BDD100K dataset, it is composed of the three classes: pole, traffic sign and traffic light), as illustrated in Figure 2.7. From each cluster, we can estimate the visible pole base. To mitigate wrong class prediction, we include a minimum height requirement in pixels and a maximum pixel threshold to prevent the annotation of excessively large areas erroneously classified as poles.

As visible in the resulting image of the Figure 2.7, due to a larger definition of poles, some bollards are annotated by this approach and cannot be avoided. Besides, due to segmentation errors some poles can be missed. However, using the given approach, more poles can be annotated, especially at a large distance from the camera, where the map-based approach is limited due to the lidar refinement choice. Besides, it could help annotate unmapped poles. Seeing this result, it is clear that the two sets of annotations (from map-based and from segmentation-based) obtained are different and, depending on the fusion strategy applied, different annotation performance, more or less compliant with the final detection objectives, will be obtained.

2.4.2 *Lidar semantic segmentation*

A similar approach can be applied to lidar point clouds in order to segment the poles. In this thesis, we use the Cylinder3D network [Zhu et al., 2020] for lidar point clouds semantic segmentation. An overview of the lidar-based annotation method for the images is visible in the Figure 2.8.

All the lidar points classified as poles are first clustered into single entities. The clusters are then assigned a single 3D coordinates by taking the mean value along the x-y coordinates and the lower value of the z-coordinates. The z-coordinate is used to check whether the cluster corresponds to pole standing on the ground. For this step, we use the same method to extract the ground points ${}^{L}\mathcal{G}$ from the lidar as in the previous section. Note that we could use the results from the whole semantic segmentation of the point clouds, but it was less reliable than the proposed ground segmentation method used for map-based automatic annotation.

A pole cluster is considered as standing on the ground if there is a point in ${}^{L}\mathcal{G}$ close enough to the cluster coordinate. Finally, this point is projected onto the image to serve as an annotation.

2.4.3 *Multi-modal annotation in an object detection problem*

The goal is now to combine the annotations computed from several methods. For a given method k, the set of annotations for an image i is defined as

$$ {}^{(k)}a^{i} = \left\{ {}^{(k)}a^{i}_{j} \middle| j = 1, \dots, {}^{(k)}n^{i} \right\}, \tag{2.11} $$

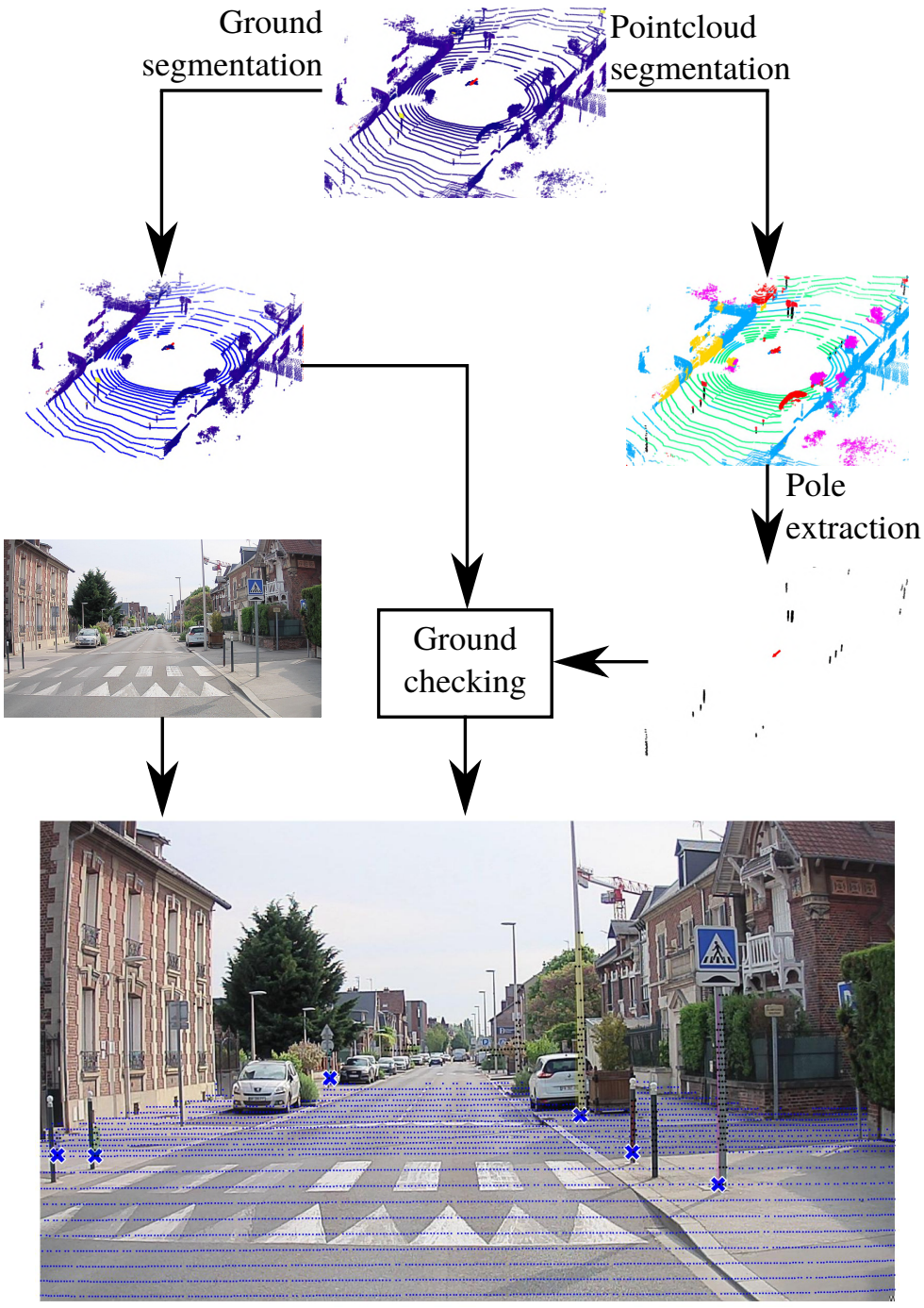


Figure 2.8: *Annotation of pole bases in images using exclusively lidar data. The original point cloud is segmented to extract pole points and generate clusters of pole points. For each cluster, the lowest point is identified to determine if the pole base is visible, ensuring that it is located on the ground. To address precision concerns with point cloud segmentation for the ground class, the ground segmentation method proposed by [Jiménez et al., 2021] is used instead of relying solely on segmented points identified as ground. Points identified as pole bases are then projected onto the image and represented by annotations highlighted with blue crosses. For visualization purposes, ground points and pole points are projected onto the image and distinguished by blue and black points, respectively.*

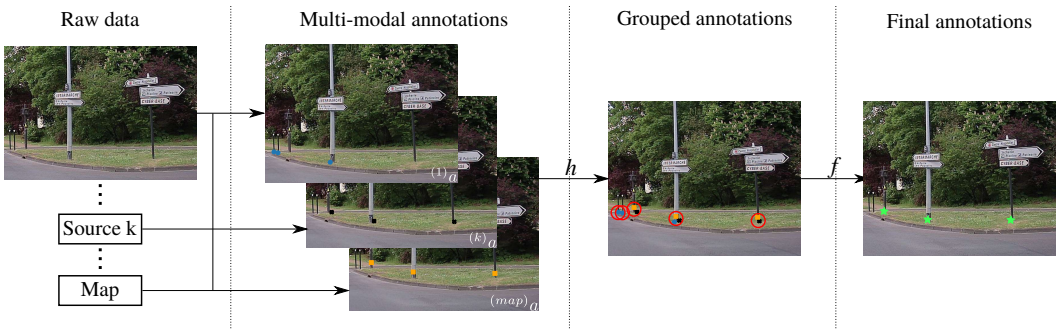


Figure 2.9: *Steps in an automatic multi-modal labeling method. Multiple annotation sets are obtained from the images and diverse data sources, including the vector map. Thanks to a data association function h, annotations are grouped to derive final annotations through a fusion function f. The final annotations are displayed with green stars in the last image.*

where ${}^{(k)}n^i$ is the number of detected objects in image i by the method k and ${}^{(k)}a^i_j$ encodes an object annotation, which is here the coordinates of a single point $\left(u^i_j, v^i_j\right)$. For N images, the resulting annotation set by method k is

$$ {}^{(k)}a = \left\{ {}^{(k)}a^i \,\middle|\, i = 1, \ldots, N \right\}. \tag{2.12} $$

In [Noizet et al., 2024], we proposed the entire annotation process visible in Figure 2.9.

The fusion process of K annotation sets coming from K independent methods can be decomposed into two principal steps.

Clustering of annotations

The first step is to define a data association function h that, given a set of annotations $A = \left\{ {}^{(1)}a^i, {}^{(2)}a^i, \ldots, {}^{(K)}a^i \right\}$, returns a set of clusters of annotations corresponding to the annotation of the same element along different modalities:

$$ h(A) = \left\{ c^i_1, c^i_2, \ldots, c^i_M \right\} \tag{2.13} $$

where M is the number of different annotated elements and each c^i_j is a set containing at most one element of each ${}^{(k)}a^i$:

$$ c^i_j = \begin{cases} \left\{ {}^{(k)}a^i_\ell \right\} & \text{if a pole is detected only by the method } k \\ \left\{ {}^{(k)}a^i_\ell, {}^{(k')}a^i_{\ell'} \right\} & \text{if detected only by the methods } k \text{ and } k' \\ \vdots & \\ \left\{ {}^{(1)}a^i_{\ell_1}, \ldots, {}^{(K)}a^i_{\ell_K} \right\} & \text{if detected by all the methods} \end{cases} \tag{2.14} $$

Note that $h(A)$ is a partition of the sets in A:

$$\forall j \neq \ell,\ c_j^i \cap c_\ell^i = \emptyset, \quad \text{and} \bigcup_{j=1,\dots,M} c_j^i = \bigcup_{k=1,\dots,K} {}^{(k)}a^i \tag{2.15}$$

In our case, because there are no explicit semantic classes encoded in the annotations, the criteria used in h is typically based on geometric proximity metrics, *e.g.* Euclidean distance between points.

The second step is to create a fusion function f that combines annotations from a set c_j^i into one. In our case, this means determining the specific location in the image for the corresponding object. There are various approaches available. One can calculate the average point of annotations, or alternatively, choose the best one based on a quality or confidence metric associated with the annotations.

From these two functions f and h, we can define, for image i, some consensus annotation sets defined as

$${}^{(1:K)}_{q}a^i = \{f(C)\,|C \in h(A) \text{ and } |C| \geqslant q\} \tag{2.16}$$

where $|C|$ is the cardinality of the set of annotations C and $q \in \{1,\dots,K\}$ is a degree of consensus. The set ${}^{(1:K)}_{q}a^i$ corresponds to the fused annotations of all the objects that have been annotated by at least q methods among the K ones. Two particular sets ${}^{(1:K)}_{1}a^i$ and ${}^{(1:K)}_{K}a^i$ corresponds to the union and the intersection of the annotations sets, respectively. In other words, if an annotation belongs to ${}^{(1:K)}_{1}a^i$, it means that at least one method agrees with it while if it belongs to ${}^{(1:K)}_{K}a^i$ then all the K methods agree.

2.4.4 *Fusion properties*

Depending on the fusion strategy, an increase of precision or recall of the automatic annotation is expected. Let ${}^{(k)}r$ be the recall of the k-th method defined as

$${}^{(k)}r = \mathbb{P}\left(Z_k = 1 | X = 1\right), \tag{2.17}$$

and ${}^{(k)}p$ be the precision of the k-th method, defined as

$${}^{(k)}p = \mathbb{P}\left(X = 1 | Z_k = 1\right). \tag{2.18}$$

where X and Z_k are binary random variables. The first encodes the existence of an object and the second if it has been detected by the k-th method.

For simplicity, $\mathbb{P}\left(Z_k = 1 | X = 1\right)$ and $\mathbb{P}\left(X = 1 | Z_k = 1\right)$ are noted $\mathbb{P}\left(Z_k | X\right)$ and $\mathbb{P}\left(X | Z_k\right)$ in the following.

By applying the union of annotations, the resulting recall is guaranteed to increase since it provides more annotations. However, a decrease in precision may occur if false negatives are added. In fact, the recall of the union of K independent methods is:

$$
\begin{aligned}
{}^{(1:K)}_{1}r &= \mathbb{P}\left(Z_1 \text{ or } \cdots \text{ or } Z_K | X\right) \\
&= 1 - \mathbb{P}\left(\text{not } Z_1 \text{ and } \cdots \text{ and not } Z_K | X\right)
\end{aligned}
\tag{2.19}
$$

By using conditional independence, one can show that

$$
{}^{(1:K)}_{1}r = 1 - \underbrace{\prod_{k=1}^{K}\left(1 - {}^{(k)}r\right)}_{\leqslant \max_k 1-{}^{(k)}r} \geqslant \max_k {}^{(k)}r. \tag{2.20}
$$

The intersection of annotations can lead to a precision improvement, though not guaranteed, but inevitably decreases the recall since it removes some annotations. The precision of the intersection of K independent methods

$$
{}^{(1:K)}_{K}p = \mathbb{P}\left(X | Z_1 \text{ and } \cdots \text{ and } Z_K\right), \tag{2.21}
$$

can be written as follows by using Bayes' rules:

$$
\begin{aligned}
{}^{(1:K)}_{K}p &= \frac{\mathbb{P}\left(Z_1 \text{ and } \cdots \text{ and } Z_K | X\right)\mathbb{P}\left(X\right)}{\mathbb{P}\left(Z_1 \text{ and } \cdots \text{ and } Z_K\right)} && (2.22)\\
&= \frac{\mathbb{P}\left(X\right)}{\prod_{k=1}^{K}\mathbb{P}\left(Z_k\right)}\prod_{k=1}^{K}\mathbb{P}\left(Z_k | X\right) && (2.23)\\
&= \frac{\mathbb{P}\left(X\right)}{\prod_{k=1}^{K}\mathbb{P}\left(Z_k\right)}\prod_{k=1}^{K}\frac{\mathbb{P}\left(X | Z_k\right)\mathbb{P}\left(Z_k\right)}{\mathbb{P}\left(X\right)} && (2.24)\\
&= \frac{1}{\mathbb{P}\left(X\right)^{K-1}}\prod_{k=1}^{K}{}^{(k)}p && (2.25)
\end{aligned}
$$

where $\mathbb{P}(X)$ is the prior probability that a sample X corresponds to a real object. By considering the negation of X, i.e., $\neg X \Leftrightarrow X = 0$, we have

$$
\begin{aligned}
1 - {}^{(1:K)}_{K}p &= \mathbb{P}\left(\neg X | Z_1 \text{ and } \cdots \text{ and } Z_K\right) && (2.26)\\
&= \frac{1}{\left(1 - \mathbb{P}\left(X\right)\right)^{K-1}}\prod_{k=1}^{K}\left(1 - {}^{(k)}p\right) && (2.27)
\end{aligned}
$$

By rewriting ${}^{(1:K)}_{K}p$ as follows

$$
{}^{(1:K)}_{K}p = \frac{{}^{(1:K)}_{K}p}{{}^{(1:K)}_{K}p + \left(1 - {}^{(1:K)}_{K}p\right)} \tag{2.28}
$$

and replacing the terms ${}^{(1:K)}_{K}p$ by Eq. (2.25) and $\left(1 - {}^{(1:K)}_{K}p\right)$ by Eq. (2.27) we obtain

$$ {}^{(1:K)}_{K}p = \frac{\prod_{k=1}^{K} {}^{(k)}p}{\prod_{k=1}^{K} {}^{(k)}p + \left[\frac{\mathbb{P}(X)}{1-\mathbb{P}(X)}\right]^{K-1} \prod_{k=1}^{K} \left(1 - {}^{(k)}p\right)} \tag{2.29} $$

In the general case, the precision from Eq. (2.29) is not always higher than the individual precisions. A trivial example is when one of the methods has a precision ${}^{(k)}p = 0$ then the precision of the intersection goes down to zero.

In the case where none of the methods has a zero precision, Eq. (2.29) can be rewritten as

$$ {}^{(1:K)}_{K}p = \frac{1}{1 + \frac{1-\mathbb{P}(X)}{\mathbb{P}(X)} \left[\frac{\mathbb{P}(X)}{1-\mathbb{P}(X)}\right]^{K} \prod_{k=1}^{K} \frac{1-{}^{(k)}p}{{}^{(k)}p}} \tag{2.30} $$

$$ = \frac{1}{1 + \frac{1-\mathbb{P}(X)}{\mathbb{P}(X)} \prod_{k=1}^{K} \frac{\mathbb{P}(X)}{{}^{(k)}p} \frac{1-{}^{(k)}p}{1-\mathbb{P}(X)}}. \tag{2.31} $$

Under the hypothesis that the precision ${}^{(k)}p$ of each method is higher than the prior $\mathbb{P}(X)$, *i.e.*, better than random, then we have:

$$ \forall k \in \{1, \ldots, K\}, {}^{(k)}p \geqslant \mathbb{P}(X) \tag{2.32} $$

$$ \Rightarrow \prod_{k=1}^{K} \underbrace{\frac{\mathbb{P}(X)}{{}^{(k)}p}}_{\leqslant 1} \underbrace{\frac{1-{}^{(k)}p}{1-\mathbb{P}(X)}}_{\leqslant 1} \leqslant \max_{k} \frac{\mathbb{P}(X)}{{}^{(k)}p} \frac{1-{}^{(k)}p}{1-\mathbb{P}(X)} \tag{2.33} $$

$$ \Rightarrow \frac{1-\mathbb{P}(X)}{\mathbb{P}(X)} \prod_{k=1}^{K} \frac{\mathbb{P}(X)}{{}^{(k)}p} \frac{1-{}^{(k)}p}{1-\mathbb{P}(X)} \leqslant \max_{k} \frac{1}{{}^{(k)}p} - 1 \tag{2.34} $$

$$ \Rightarrow {}^{(1:K)}_{K}p \geqslant \max_{k} {}^{(k)}p. \tag{2.35} $$

Therefore, the precision of the intersection of independent methods increases only when the individual methods have high enough precision.

However, if none of the methods has a zero precision, *e.g.*, $\prod_{k=1}^{K} {}^{(k)}p > 0$ and if the prior is uniform, *e.g.*, $\mathbb{P}(X) = 1/2$ then the precision is guaranteed to increase:

$$ {}^{(1:K)}_{K}p = \frac{1}{1 + \prod_{k=1}^{K} \left(\frac{1}{{}^{(k)}p} - 1\right)} \geqslant \max_{k} {}^{(k)}p \tag{2.36} $$

The balance between precision and recall performance depends on the application, one can be favored over the other and conversely.

2.5 EXPERIMENTAL RESULTS

2.5.1 *Dataset*

To the best of our knowledge, there is no public dataset that includes an HD map with georeferenced poles along with image and lidar data, which are essential for our various automatic annotation approaches.

We conducted experiments using an experimental Renault ZOE equipped with a Hesai Pandora sensor integrating a 40-layer lidar with monocular cameras. The cameras and lidar are synchronized, enabling the proposed annotation strategies. To compute the reference poses needed for our map-based automatic annotation method, the vehicle was equipped with a NovAtel SPAN-CPT GNSS/IMU and localization data were post-processed with PPK computations for high-accuracy localization.

We carried out multiple data acquisitions, as explained in Appendix A, which provides more details on the experimental platform. From these sequences, we selected two that were acquired on different days under varying conditions to manually annotate 2,830 images. Given our HD map characteristics and our deliberate exclusion of short-lived objects such as bollards, we focused our annotations on elements like traffic signs, traffic lights, and streetlamps.

As a reminder, our objective is to develop specialized detectors for the elements stored in our HD map, which will be used for localization. In total, we annotated 9,017 poles in the images, representative of all our datasets and potential AD scenarios described in Section 1.1.

2.5.2 *Single annotation methods for images*

To match each automatic annotation with its corresponding manual annotation and identify true positives (TP), false positives (FP), and false negatives (FN), we used a Unique Nearest Neighbor (UNN) approach in the image space, relying on Euclidean distance to associate annotations between the two sets. This process involves finding the closest manual annotation for each automatic annotation. If multiple automatic annotations are matched to the same manual annotation, only the nearest one is counted as a true positive, while the others are marked as false positives. A maximum distance threshold was applied to prevent associations with overly distant manual annotations and accurately identify false positives.

We evaluate the automatic annotation methods developed in terms of precision and recall. The results are reported in Table 2.1. S refers to the image segmentation-based annotation method, L to the lidar segmentation-based approach, and M to the map-based one.

S generates approximately five times as many pole base candidates as the other two methods. This method is the most generic, annotating anything considered as a pole by the semantic segmentation network, resulting in the highest recall. However, its precision is much lower than M due to its generality. M, although

Table 2.1: *Annotation evaluation of the three basic methods. Number: number of annotated poles; FP: false positive; TP: true positive; FN: false negative; Prec: precision (%); Rec: recall (%); MAE-x: median absolute error in pixels along the x-axis.*

Method	Number	FP	TP	FN	Prec.	Rec.	MAE-x
M	3830	**589**	3241	5776	**84.6**	35.9	3.91
S	18084	10808	**7276**	**1741**	40.2	**80.7**	**1.00**
L	3266	1221	2045	6972	62.6	22.7	2.63

annotating fewer pole bases, achieves the best precision since our manual annotations correspond to the classes contained in the map. Because the map-based method relies on the lidar data, the recall gets limited by the range of the sensor and the sparsity of the point cloud. L exhibits a higher precision than S, indicating fewer point cloud segmentation errors compared to image segmentation errors in our case. Moreover, it is simpler to extract poles from segmented point clouds than from segmented images. However, similarly to M the recall of L is lower due to the lidar sensor limitation.

To accurately detect pole bases, our goal is to minimize horizontal positioning errors in the image frame. In fact, we suppose that errors along the y-coordinate are less impactful, given the vertical nature of pole objects and the use of the detection in a localization context. The median absolute error in Table 2.1 shows that S has a pixel-level error, L has an error twice as large and the error for M is four times higher. The histograms of horizontal errors depicted in Figure 2.10 shows that M is more prone to higher errors, often exceeding 5 pixels and occasionally reaching up to 10 pixels, compared to other approaches. It can be explained by the fact that only segmentation errors can occur from the method S. Errors due to sensor calibration, segmentation and clustering can impact the method L. Finally, errors due to vehicle positioning errors, sensor calibration, map and ground segmentation can arise from the method M.

Examples of automatic annotation results from [Noizet et al., 2024] are visible in Figure 2.11. It illustrates that none of the methods are capable of annotating all poles visible in the image, and S incorrectly labels multiple bollards as poles, highlighting the need for a multi-modal approach to create a better annotation set.

2.5.3 *Annotation association and fusion for image automatic annotation*

For the association of the annotations for the multi-modal approach introduced in Section 2.4, we use the same UNN approach as used for the evaluation of single methods. The final association function $\mathfrak{h}$ with three modalities consists in grouping all the pairwise associations between two sets of annotations provided by two different methods. Once a set C of annotations is obtained, the fused annotation

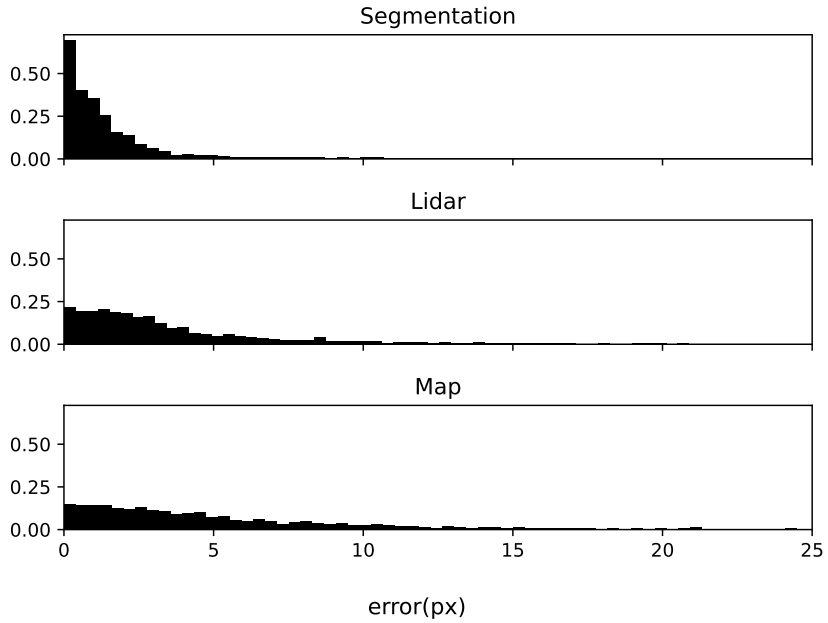


Figure 2.10: *Histograms of absolute horizontal positioning errors between automatic annotations and manual annotations.*

$f(C)$ is defined as the best annotation contained in C considering a preference order: $S \succ L \succ M$. This order is determined by the presence of potential positioning error sources identified.

We assess various fusion strategies as previously defined, testing unions and intersections of sets. To build the pole base detector aligned with the HD map, we specifically consider combinations involving M as our objective is to detect elements present in the map for localization. We exclude combinations obtained only from L and S, as they may annotate pole structures as bollards, which are not accounted for in the map. The results are presented in Table 2.2.

Table 2.2: *Annotation evaluation of the possible fusion strategies. "|" and "&" indicate respectively union and intersection of annotations. Number: number of annotated poles; FP: false positive; TP: true positive; FN: false negative; Prec: precision (%); Rec: recall (%).*

Method	Number	FP	TP	FN	Prec.	Rec.
M | L	5609	1787	3822	5195	68.1	42.4
M | S	18695	11204	7491	1526	40.1	83.1
M | S | L	19400	11830	**7570**	**1447**	39.0	**83.9**
M & L	1487	41	1446	7571	97.2	16.0
M & S	3219	213	3006	6011	93.4	33.3
M & S & L	1362	**15**	1347	7670	**98.9**	14.9

(a) *M annotation.*

(b) *S annotation.*

(c) *L annotation*

Figure 2.11: *Examples of automatic annotations obtained using three different methods from [Noizet et al., 2024]. They are depicted with blue crosses. Green circles represent reference annotations defined by humans and correctly annotated automatically. The red ones are those that are missed.*

As expected, the union of annotation sets improves the recall. The recall is limited (less than 45%) when S is not involved. However, S decreases strongly the precision of the union by almost one half.

Conversely, applying the intersection significantly improves precision to more than 90%. Here, despite the poor S results in terms of precision (around 40%), it does not negatively impact the precision of intersections. However, the intersection drastically reduces the number of annotations and the recall, even if the best precision is obtained when involving all methods for a small recall. M & S is the best intersection of sets possible to guarantee the highest recall possible with a high precision.

Combinations involving at least M and S seem to be the best candidates for annotation improvement. The union of at least M and S provides a very high recall, indicating that few pole bases are missed. On the other hand, the intersection ensures very high precision, thus limiting false candidates.

As a conclusion, each method and combination provides different annotation performance. The recall is maximized by M | S | L, while the precision is maximized by M & S & L. Each set will probably impact differently the detector training and the overall performance could be deeply impacted. However, when involving at least M and S we are capable of identifying precise annotations doing the intersection of sets and the union of sets can help us identify ambiguous cases, where only one method was capable of annotating the corresponding pole bases, whose some correspond to the map annotation errors we want to mitigate. Consequently, it could be interesting to identify the overall impact of each annotation approach and combination on learning and probably develop a strategy to manage identified ambiguous pole base cases.

2.6 CONCLUSION

To build a map-aligned pole detector, annotated datasets respecting the pole definition used in the vector map are essential. Map-based automatic annotation can be employed for this purpose. However, due to various limitations, such as map errors and the annotation process itself, incorrect annotations may be added, and some poles may be missed. Both types of errors can significantly impact performance by providing false examples of poles or non-poles to the detector employed.

To address these issues, multi-modal automatic annotation approaches that use all available data can be developed. Annotations are extracted from maps, lidar data, and images, then fused to create an enhanced annotation set based on a chosen combination strategy. Depending on the combination applied, the precision or recall of the overall annotation can be improved. Intersection and union of sets respectively enhance precision and recall. These automatic annotation techniques, in combination with map-based annotations, enable the generation of training data tailored to the specific characteristics of the mapped environment, thereby ensuring detectors are optimized for real-world deployment respecting the ODD.

Furthermore, this approach enhances adaptability, allowing for easy adjustment to accommodate new detectors, changing environments, and evolving requirements.

However, depending on the strategy applied, achieving high precision or high recall is possible, but not both simultaneously, leading to sets that inevitably contain either incorrect annotations or missed pole bases. Introducing additional sources or refined annotation methods could help improve overall performance, despite potentially increasing the number of possible combinations and complicating the analysis. For example, using the same data, multiple image segmentation techniques can be employed, to generate several segmentation-based annotations.

However, errors, even if potentially reduced, will still occur. This is primarily because the map is unable to annotate unmapped poles, while other methods may annotate objects that resemble poles but are not included in the map. Their impact on the learning of proposed detectors will be analyzed in the next chapter, and, using different combinations, we can try to identify and manage the given issues.

Chapter 3

Training Pole-like Feature Detectors with Automatic Annotations for Map-Based Localization

Contents

3.1 Introduction

Using the data introduced in the previous chapter, we obtained a set of annotated images suitable for object detection training. Depending on the object detection

method chosen, an appropriate data representation must be defined. Besides, multiple sets of annotations were obtained from different sources and the final annotation set must be constructed thoughtfully, potentially using all available sets.

Any chosen annotation approach may result in missing or incorrect annotations in the image. In a single image, multiple pole bases may need to be detected and should be accurately annotated if they are present. Therefore, additional data manipulation is necessary to ensure the best completeness and accuracy of the annotation sets given the images used.

In this chapter, we propose to adapt well-known object detection methods using bounding boxes for the specific task of detecting pole bases. In our case, the bounding box provides the necessary visual context for identifying pole bases. Initially, we train a detector using map-based automatic annotations to evaluate the feasibility of this approach and to determine an optimal box size. To enhance performance, we then explore training with various combinations of automatic annotation modalities.

Additionally, we explore an approach that uses image editing with black patches to address multi-modal annotation uncertainty, aiming to further improve overall detection performance.

Since the achievable performance and the optimal box size are influenced by the quality of the training annotation set, we also investigate how different types of annotation errors impact both detection performance and the selection of bounding box size.

The results are experimentally validated on the same dataset used in Chapter 2, leveraging manual annotations. However, given our capability to perform automatic annotation, we also carry out an evaluation on automatically annotated data to study the usability for validation.

3.2 OBJECT DETECTION WITH NEURAL NETWORKS: A STATE-OF-THE-ART

For cameras, countless models have been developed by researchers to address specific detection needs. Given the map elements we aim to detect here, our primary focus lies on object localization, object classification, object detection, and image segmentation. In object localization, the model is trained to detect desired objects using bounding boxes. This requires datasets containing images labeled with bounding boxes that specify the location of objects. Object classification involves training the model to assign a class to a given image of an object. This process relies on datasets comprising images of various objects, each accompanied by corresponding class labels. Object detection aims to accomplish both localization and classification tasks. Most of the models employ Convolutional Neural Networks (CNNs). Early object detection models, such as R-CNN [Girshick et al., 2014] and its variants, followed a two-step process: localization followed by classification.

Nowadays, the most widespread approach for object detection consists in single-stage approaches as You Only Look Once (YOLO) models [Wang et al.,

Figure 3.1: *Traffic signs detected by a YOLOv7 neural network [Wang et al., 2022] trained on BelgiumTS dataset [Timofte et al., 2014]*

2022]. These models directly predict bounding boxes and class probabilities from a single pass through the network, making them faster and more efficient for real-time applications. An example of traffic sign detections from a YOLOv7 neural network is visible in Figure 3.1.

Semantic segmentation models have the disadvantage of being more computationally intensive and providing extensive data, which make the extraction of objects of interest longer than for an object detection method. A balance between object detection and image segmentation, instance segmentation, detects objects using bounding boxes and then precisely segments the instances. Additional datasets are necessary for training but can also serve for landmark detection. For instance, [Barbosa et al., 2021] developed a pole detector trained on a custom dataset using an R-CNN variant proposed by [He et al., 2017] for instance segmentation.

In our case, due to the limitations of the available annotations, we are constrained to using object detection approaches. A challenge arises from the wide range of available models today. While well-known models, as YOLO, are well-documented, choosing the right one among the current flood of options can be challenging, typically in our case, where initially we only have annotations as points. In fact, to the best of our knowledge, no existing neural network is designed specifically for detecting only points in images.

Besides, our annotation sets are obtained through automatic processing and contain errors and even though deep learning models may exhibit a small degree of tolerance for annotation errors in object detection contexts, the accumulation of numerous errors can significantly degrade the network performance [Chachuła et al., 2023; Chadwick et al., 2019]. Particular attention should therefore be given

to the quality of annotations. The sources of performance degradation are missed annotations, unwanted added annotations and incorrectly positioned annotations.

To account for these errors, one can try to further clean out the data, or instead manage and quantify the uncertainties. For instance, confident learning involves characterizing label quality using the model to prune errors from training sets [Chachuła et al., 2023; Northcutt et al., 2021]. However, since the model is employed to identify errors, it seems challenging to prune labels corresponding to false positives that are similar to the objects we aim to detect. It is typically true with bollards we do not want to detect in our case.

Some methods involve modifying the loss used in the network to handle uncertainties as done by [Reed et al., 2014] and manipulating soft-labels. However, modifying the loss or the network itself limits the usability of these approaches, potentially requiring more challenging work than selecting a state-of-the-art model to learn. Additionally, estimating uncertainties is not particularly straightforward and depends on the annotation approaches used.

Therefore, in our case, we propose adapting widely used state-of-the-art detectors, such as YOLOv7, as we did in [Missaoui et al., 2023] instead of experimenting with lesser-known networks or modifying existing ones. YOLOv7 have multiple advantages as its excellent real-time performance, user-friendly nature, and extensive documentation. Furthermore, since our annotation methods may introduce errors, we aim to develop an approach that effectively manages ambiguous annotation cases without altering the network or relying on network-specific strategies if possible.

3.3 FROM POINTWISE ANNOTATIONS TO BOUNDING BOXES

YOLOv7 network and any other object detection network use bounding boxes approaches. They are used to contain the entire object detected and help train a representation of the given object and make it detectable. Here, we need to use bounding boxes to represent the pole base. However, unlike classical detected objects as cars, persons or traffic signs, pole bases are not clearly defined objects that could be bounded by a box. Consequently, they are used to encode the visual context.

From a set of annotations a^i as defined in Eq. (2.11) for a given image i, we can define a set of bounding boxes:

$$B^i = \left\{ B^i_j = \left(u^i_j, v^i_j, w, h\right) \middle| \, j = 1, \ldots, n^i \right\}, \tag{3.1}$$

Where $\left(u^i_j, v^i_j\right)$ the image coordinates of the annotation a^i_j corresponding to the center of the box, w the width of the box and h the height of the box, two tunable parameters. To simplify the process, we choose to set the same width and height for all annotations, even if their distance from the camera are different and consequently their representation in the image. The farther they are from the camera, the smaller they appear in the image.

Figure 3.2: *Example of bounding box tuning for pole base detection. The chosen box size significantly impacts the ability to accurately distinguish closely spaced pole bases.*

The main question here is whether it is feasible to build a pole base detector from such annotations. If it is, we must then identify the necessary size of the context and the factors that influence this size.

An example of varying box sizes for the same image is illustrated in Figure 3.2. Small contexts may pose challenges for detecting pole bases, particularly considering annotation positioning inaccuracies discussed in previous chapters. In fact, they may fail to adequately represent the pole bases, depending on the annotation sources used. On the other hand, large contexts may result in a loss of detection capability, with boxes overlapping quickly, and potentially including multiple close poles.

When poles are very close to each other, as shown here, there is concern whether the network will be able to detect both. Additionally, due to the functioning of YOLOv7, even if, in a first step, the network successfully identifies the two poles, the final detection results may only include one of the two bounding boxes.

In fact, as output, the detector generates sets of bounding boxes corresponding to pole bases candidates with confidence information. Typically, for an image i, we have

$$D^i = \left\{ D^i_j = \left(u^i_j, v^i_j, w^i_j, h^i_j, s^i_j \right) \middle| \, j = 1, \ldots, m^i \right\}, \tag{3.2}$$

Where $\left(u^i_j, v^i_j \right)$ represents the image coordinates of a pole base candidate D^i_j corresponding to the center of the box. The variables w^i_j and h^i_j denote the dimensions of the box, the width and the height respectively, potentially different from the one used for learning. Each candidate is assigned a confidence score s^i_j ranging

(a) *Before NMS.* ***(b)*** *After NMS.*

Figure 3.3: *Example of removal of a bounding box corresponding to a real pole base due to NMS step.*

from 0 to 1, typically representing a combination of a positional accuracy score and an object class score in a multi-class context.

Once trained, when an image is passed through the network, it first generates a large number of bounding boxes. Numerous of these boxes do not correspond to the objects it should detect and have very low scores. Moreover, numerous boxes correspond to the same object but show significantly varying scores.

Given the excessive number of proposed boxes, and the need to avoid multiple boxes for the same objects, two techniques are employed to filter and reduce the pole base candidates after the initial detections are generated.

Firstly, the Non-Maximum Suppression (NMS) phase eliminates redundant bounding boxes around the same object. It selects the box with the highest score and removes others that significantly overlap with it. This process ensures that each detected object is represented by a single, optimal bounding box.

To evaluate the overlap between two detections D_j^i and D_k^i, the Intersection-over-Union (IoU) is used:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}} \tag{3.3}$$

An IoU score ranges from 0 to 1. A score of 0 indicates that D_j^i is a different object than D_k^i. A score of 1 indicates perfect overlap, meaning the boxes correspond exactly to the same object. To consider if two detections overlap or not, a IoU threshold is defined for the NMS step.

NMS could be problematic in scenarios like the one shown in Figure 3.3, where two boxes representing different poles should both be kept, and only two redundant boxes should be removed. However, due to the IoU threshold used, only one bounding box is kept.

Then, after applying NMS, we can further filter the remaining boxes by retaining only those with scores above a certain threshold, thereby excluding detections that are unlikely to represent a pole base.

3.4 OBJECT DETECTION EVALUATION: PRECISION-RECALL CURVES AND OTHER METRICS

Using a validation set of multiple annotated images, we can evaluate the performance of the detection network. For that, we need to identify true positive candidates and wrongly detected poles, the false positives. We use the method classically used in an object detection context with bounding boxes by computing the IoU for all the combinations from the set of annotated bounding boxes B^i and the set of detected boxes D^i.

First, detections are sorted in descending order of their scores. For each detection, following this order, the annotation with the highest IoU is identified. If this annotation has not yet been associated with any detection, the detection is considered a true positive and is matched to that annotation. If all annotations have already been associated with detections, the detection is considered a false positive.

Thus, for the complete validation set, we can identify true positives and false positives, allowing us to calculate the precision and recall achieved. However, if calculated at this stage, we would consider all detections, despite their confidence scores potentially providing valuable insights in the probability of existence of a real pole base, leading to poor performance. Detections with scores below a specified threshold $\mathcal{T}_s$ are often disregarded as highly improbable. This selection approach can significantly modify the results. Removing detections with low scores tends to reduce false positives significantly, albeit at the cost of potentially removing a few true positives.

Then, as depicted in Figure 3.4, we can obtain multiple precision and recall pairs across different thresholds $\mathcal{T}_s$, typically set to scores observed on the validation set. These pairs form a Precision-Recall (PR) curve. Higher thresholds generally yield high precision at the expense of lower recall, whereas lower thresholds yield higher recall with potentially lower precision.

As depicted in the figure, the optimal PR curve in red is obtained when decreasing the score threshold function only adds true positives before adding all false positives, firstly going from a recall of 0 to 1 with a precision of 1 then reducing the precision to 0 for a recall of 1. Additionally, achieving a recall of 1 indicates that the detector is capable of identifying all pole bases.

The effectiveness of learning is reflected in how closely the PR curve approaches the optimal one. If one curve always lies below others (as the black one), it indicates that the learned detector is significantly outperformed by others. When one curve intersects another at a specific point corresponding to a threshold $\mathcal{T}^*$, one detector can surpass the other depending on the value of the threshold. For example the detector corresponding to the green curve surpasses the one corresponding to the blue curve until a given threshold $\mathcal{T}_s^*$. For any threshold $\mathcal{T}_s^k < \mathcal{T}_s^*$, the other detector surpasses this one. Note that none of these three curves achieves a recall of 1, as it is common in object detection tasks that not all objects are detected.

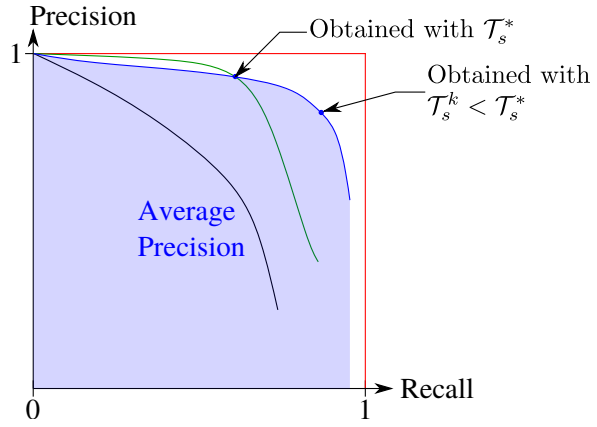


Figure 3.4: *Examples of precision-recall (PR) curves. Each point represents the precision and recall pair for a given threshold. PR curves tend to decrease as the threshold is lowered, resulting in lower precision and higher recall. The red curve represents the optimal PR curve. The black curve represents a detector that is outperformed by others. The green and blue curves correspond to two detectors with different performance, where the green detector outperforms the blue detector for thresholds higher than* $\mathcal{T}_s^*$.

The performance highlighted by a curve can be summarized using one scalar value: the average precision (AP). It corresponds to the area below the PR curve as depicted for the blue curve in the Figure.

$$\mathrm{AP} = \int_{r=0}^{1} p(r)\,\mathrm{d}r \tag{3.4}$$

where r is the recall and $p(r)$ the precision for a given recall. In practice, the PR curve is constructed using n points representing precision/recall pairs obtained by applying each observed score s_k, $k = 1 \ldots n$, sorted in descending order, from the validation set as a threshold. AP is then approximated by the following formula:

$$\mathrm{AP} = \sum_{k=1}^{n} \max\left(p_k, p_{k-1}\right) \cdot |r_k - r_{k-1}| \tag{3.5}$$

where p_k and r_k are the precision and recall obtained with the threshold s_k, respectively. In this thesis, we evaluate our pole base detectors for images using the PR curves and APs.

Our ultimate goal is to use this detector for localization purpose. Consequently, it is essential that the detector minimizes false positives, accurately identifies numerous pole bases, and positions them accurately with an acceptable positioning error for the center of the bounding box. To better evaluate the accuracy of the bounding box centers, we measure the horizontal distance relative to the point-wise annotations. This focus on horizontal error is due to the fact that, for vertical structures like poles, the horizontal positioning error is more critical.

3.5 MAP-BASED POLE BASE DETECTION LEARNING

As a reminder, our goal is to develop a pole base detector capable of specifically identifying those provided by an HD map. Exploring the feasibility of constructing such a detector is essential.

Previously, we introduced a method to extract automatically annotations from the HD map. Building on this, we aim to determine if a detector can be effectively trained and assess the achievable performance. Additionally, identifying the appropriate box size to provide sufficient context for the network during training is crucial.

3.5.1 *Dataset*

Using the same validation set described in Section 2.5.1 composed of 2830 manually annotated images, we evaluated multiple detectors trained on 5391 automatically annotated images extracted from three sequences acquired during the data acquisition campaign detailed in Appendix A.

3.5.2 *Tuning of bounding box size*

We decided to experiment pole base detection with squared boxes of various sizes ranging from 50 to 400 pixels. Employing the official implementation of YOLOv7 with default hyperparameters, we set the IoU threshold to 0.5 during the NMS step for evaluation purposes. The model was initialized with the weights from MS COCO. The training took 14h roughly per detector on a single Tesla V100 32G GPU for 300 epochs. To visualize the impact of the training, we tested the final model obtained at the last epoch on our set of manually annotated images

PR curves of all models are visible in Figure 3.5. It shows that the choice of the bounding box size depends in reality deeply on the PR trade-off we want to apply.

As pictured by the figure, 50x50 bounding boxes is insufficient to obtain interesting performance and the model is completely outperformed by other models. For other box sizes, it is more complex to analyze since all the curves intersect at multiple points. However, depending on the section of the graph observed, the dominant curve varies and may correspond to 400x400, 300x300, 200x200, or 150x150. This means that one of these box sizes is promising for training a pole detector. However, when the 300x300 size prevails, its performance is close to that of the 400x400 size, providing similar results and thus 300x300 can be disregarded.

After analyzing another criterion, AP, as summarized in Table 3.1, the 200x200 model appears to perform the best, although the models using 400x400 and 150x150 are also competitive. Since the goal is to accurately detect pole bases for localization purposes, precise centering of the detected boxes on pole bases is crucial. The horizontal positioning is particularly critical, whereas the vertical positioning is less important since poles are vertical structures. Therefore, the mean horizontal positioning errors for each box size are also provided in the

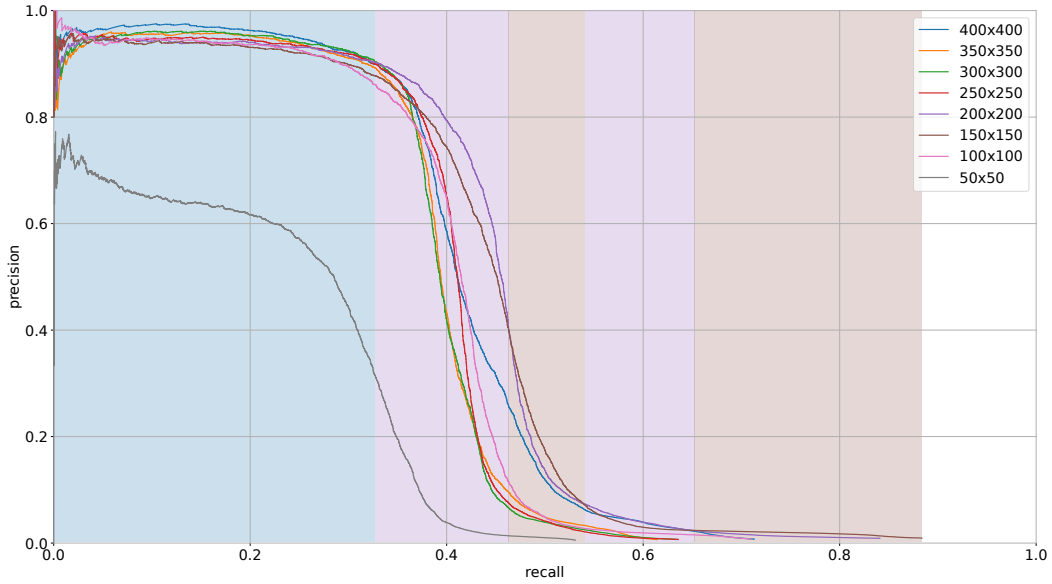


Figure 3.5: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using images automatically annotated by the map-based method. The background color indicates the predominant curve. For simplicity, when the gap between two curves is too small, no background color change is applied.*

Table 3.1: *Average precision and mean horizontal positioning errors (MAE-x) obtained after 300 epochs of training with different box sizes using images automatically annotated by the map-based method.*

Box size	50x50	100x100	150x150	200x200	250x250	300x300	350x350	400x400
AP (%)	21.2	39.2	42.5	43.2	38.9	38.3	38.4	41.3
MAE-x (px)	4.01	5.84	8.90	10.84	11.76	14.97	18.2	24.30

table. The Mean Absolute Error on horizontal axis (MAE-x) is calculated using all true positives independently of their scores.

Due to high MAE-x, 400x400 box sizes should be avoided, leading to a potential choice between 150x150 and 200x200 depending on the chosen trade-off. It is important to note that for MAE-x, all true positives are accounted for regardless of their score. Thus, applying a score threshold could also help reduce positioning errors. In this context, the metric serves as an indicator of the trend in error increase relative to box size.

Considering the slight difference between the 150x150 and 200x200 curves, it is more advantageous to prioritize the 200x200 size, as it outperforms the 150x150 across the majority of the PR space. In fact, given the overall performance decline on the PR curve for 150x150, except for a small section where it performs well, this size seems preferable mainly to minimize positioning errors.

These results collectively demonstrate the feasibility of training pole base detectors that respect the definition of the HD map, tailored to desired driving scenarios. However, their performance may be limited by the quality of automatic annotations provided by the map.

3.6 MULTI-MODAL AUTOMATIC ANNOTATION METHOD FOR LEARNING

In the Chapter 2, we proposed several automatic annotation methods and a multi-modal automatic annotation approach by associating the annotations from the different methods. We evaluated the quality of the obtained automatic annotations using the map-based annotation (M) method, as well as image segmentation-based (S) and lidar-based (L) approaches, including all possible multi-modal combinations with the map-based method.

To assess the performance of alternative automatic annotation methods for learning and the performance improvement achieved through multi-modal combinations, we propose, as previously mentioned, validating the learned detectors using the images introduced in Section 3.5. In this context, we use the previously introduced hyperparameters along with a box size of 200x200. The initial weights are the same as previously used. This study was conducted in [Noizet et al., 2024] on a smaller dataset.

The detection results obtained for all annotation methods and many combinations are summarized in Figure 3.6 by PR curves following a standard object detection evaluation.

As seen in previous section 3.5, M can achieve noteworthy precision. However, it encounters challenges in achieving higher recall, even at the sacrifice of precision. L yields to the worst results. The highest precision reachable are only for a limited recall below 10% and it exhibits a more pronounced decrease in precision as recall increases, being outperformed by M for any recall above 10%. The application of the S method for training, due to its generality, struggles to attain a precision above approximately 0.65. Overall, S achieves much higher recall than M but with a much lower precision.

For S, at low recall values indicating high confidence thresholds, the precision is lower than the maximum. This is attributed to objects with high confidence that are not considered as pole bases in our study, such as bollards. The curve rises upon including the other detections with lower confidence, yet corresponding to true positives. Besides for S and M, the beginning of the PR curves exhibits significant oscillations, indicating that at higher thresholds, true positives and false positives are being added successively.

In Figure 3.6, for different ranges of recall, the background color indicates the method with the highest precision. We can see that, although each individual method produces varying results, they are often outperformed by the combinations.

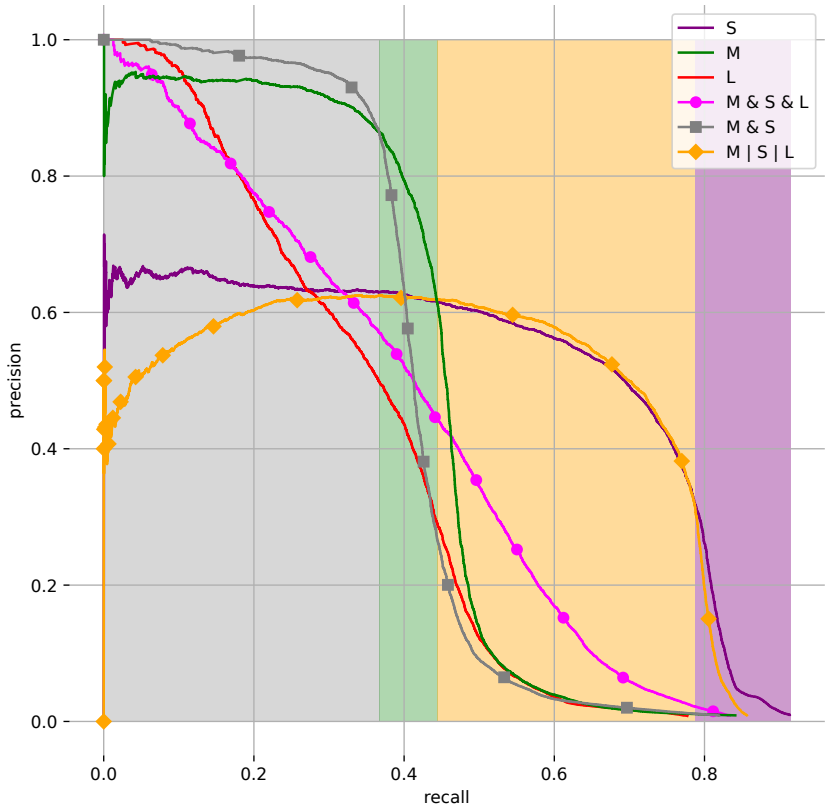


Figure 3.6: *Precision-recall curves after 300 epochs of training using different annotation approaches. The background color indicates the predominant curve.*

M & S & L provides bad results being always dominated by other models. However, M & S seems to provide interesting performance. This effect can be attributed to the significant reduction of annotations when L is intersected with any other method during automatic annotation, leading to lower recall performance. By excluding L, recall improves considerably while maintaining high precision. M & S dominates all models for a recall below 0.38 approximately with an interesting gain of precision compared to M alone. However, for higher recall objectives, M outperforms M & S illustrating the limitation of the intersection of sets to reach high recall with sufficient precision.

The union of all annotations has a similar impact to S due to the presence of all S annotations in the training set and therefore is less interesting than the other combinations. Nevertheless, for recall objectives between approximately 0.45 and 0.8, this model can surpass all the other models.

3.7 MITIGATING ANNOTATION ERRORS IN MULTI-MODAL ANNOTATION

3.7.1 *Ambiguous annotations management*

In the previous section, we demonstrated the possibility of training a pole detector with highly variable performance, irrespective of the combination used. Intersecting annotation sets enhances precision at the expense of recall, while the union of methods, due to the segmentation approach's generality, prioritizes recall over precision. These findings reveal varying annotation performances across different combinations, each influenced by specific fusion strategies.

As described in Section 2.4, diverse fused annotation sets can be generated based on the degree of consensus q. Higher values of q yield annotations with higher precision but lower recall. Therefore, for training the detector, prioritizing examples with a high degree of consensus is preferable, while disregarding those with low consensus that may inaccurately represent pole bases.

Annotation sets with the highest degree of consensus q allow us to confidently identify elements highly likely to be pole bases. Conversely, sets with lower q values help identify annotations that may not necessarily correspond to poles, highlighting ambiguous cases that can be excluded during training. However, relying solely on the set with the highest q without leveraging information from other sets may lead to limited performance as seen before.

We establish two sets: A^* comprising automatic annotations with high consensus, serving as the labels for our training set, and $\widetilde{A}$ containing all other ambiguous automatic annotations labelled by at least one method that we aim to exclude. Given a minimum consensus threshold q, we have

$$A^* = {}^{(1:K)}_{q}a \quad \text{and} \quad \widetilde{A} = {}^{(1:K)}_{1}a \Big\backslash {}^{(1:K)}_{q}a \tag{3.6}$$

Handling the ambiguous annotations $\widetilde{A}$ is not straightforward. Adding the ambiguous annotations may lead to false positive labels while removing them

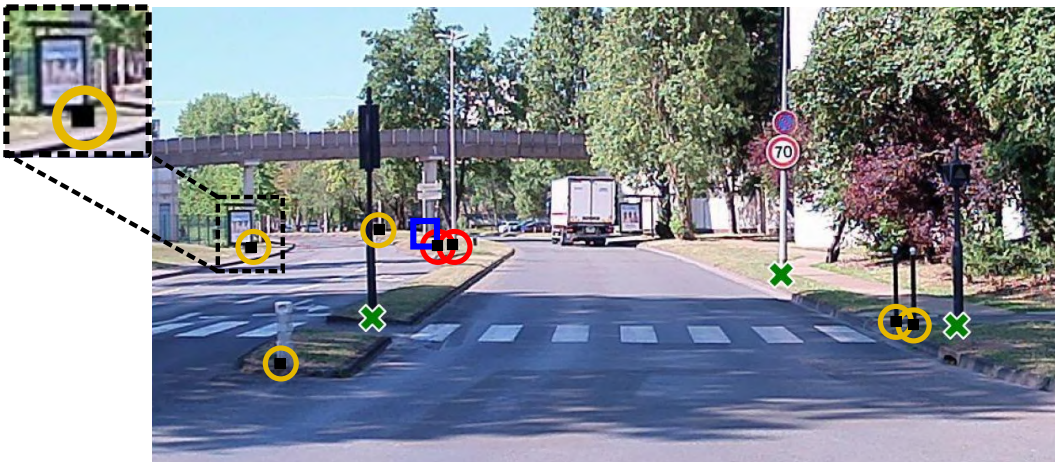


Figure 3.7: *Management of ambiguous pole bases. Green crosses: annotations with unanimous agreement. Orange circles: unnecessary black patches. Red circles: black patches to mask ambiguous pole bases. Blue square: missed pole base.*

may lead to false negative labels. In both cases, these potentially erroneous labels may lead to a decrease of performance in the training in recall or precision as previously seen with M & S and M | S | L. Then, we propose a simple bypass by adding black squared patches to mask ambiguous annotations.

Figure 3.7 illustrates such kinds of instances. In this particular case, we set $q = 3$, *i.e.*, a pole is annotated if all three modalities have annotated it as such (they are depicted with green crosses). For ambiguous annotation cases, a black patch is added to mask a part of the image corresponding to two possible situations: i) One that does not correspond to a pole base (yellow circles in Figure 3.7): while unnecessary, it is not expected to impact the model training and ii) one that actually corresponds to a pole base not included in the final label set (red circles in Figure 3.7). Adding the patch is essential to help the model during training because otherwise it leads to a false negative annotation. Even using multiple annotation methods, pole bases may still be missed, resulting in false negatives in the training set as seen with the blue square. To minimize the occurrence of such cases, the union of all methods must be able of annotating as many pole bases as possible. Simultaneously, a consensus must be established among the methods with a sufficient number of labels for training.

3.7.2 *Training improvements by handling ambiguities*

Figure 3.8 shows the results obtained when adding the black patches on the images of the training set previously used, using combinations of map-based, segmentation-based and lidar-based automatic annotations. We compared the PR curves obtained with M & S & L and M & S in Figure 3.6 with newly obtained PR curves of models trained with same combinations. For reference, the performance of model M and M | S | L are outlined in the figure.

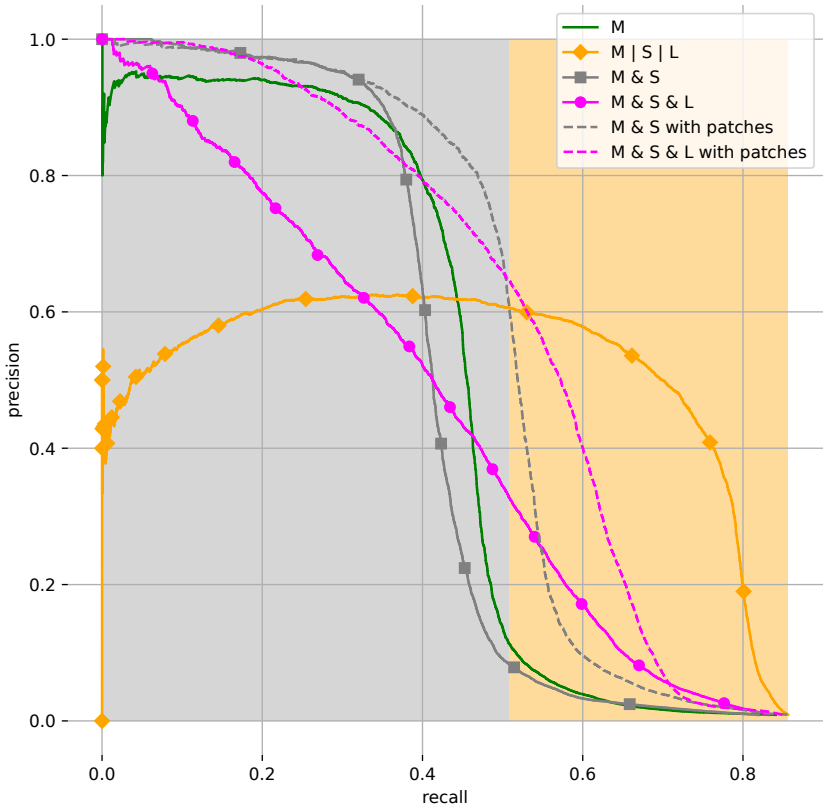


Figure 3.8: *Precision-recall curves after 300 epochs of training using black patches on images. The background color indicates the predominent curve. For simplicity, the small area corresponding to the* M & S & L *with patches is not indicated. Some curves from Figure 3.6 are kept to show the improvements.*

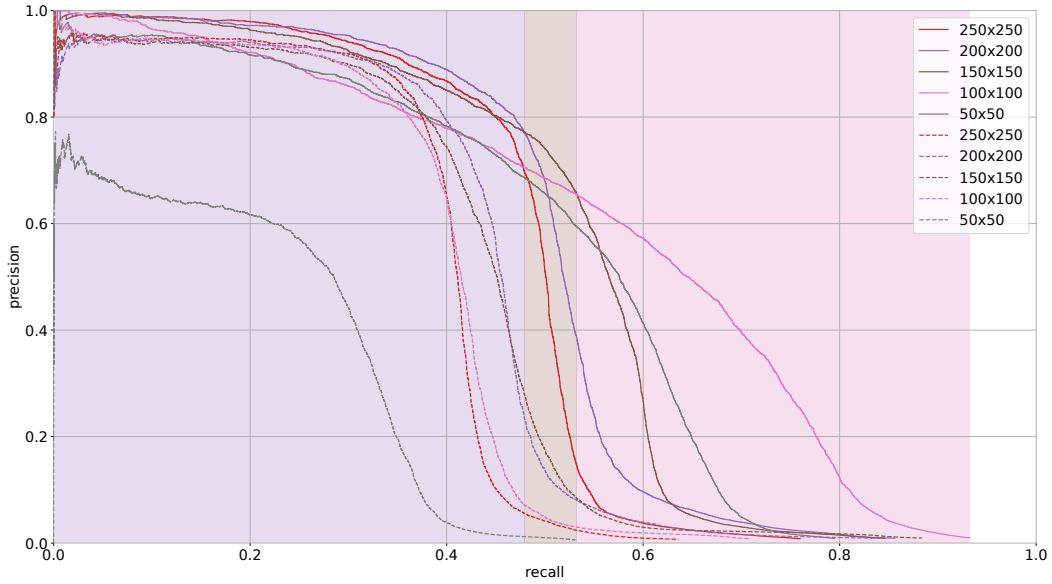


Figure 3.9: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using* M *(dashed) and* M & S *with black patches (solid) methods. The background color indicates the predominant curve. For simplicity, when the gap between two curves is too small, no background color change is applied.*

Generally speaking, adding the black patches to mask the ambiguous labels is clearly beneficial for the training of the object detector. For M & S & L , it helps improve precision and recall a lot for any score threshold. However, this model remains dominated by other models, and particularly by M & S with black patches.

Indeed, this model provides the most significant advantage, enhancing precision for recall targets above 0.32 while preserving the initial performance for lower targets. This improvement allows it to outperform M and surpass M | S | L for recall values below approximately 0.5.

We have therefore experimentally demonstrated that by combining M with S and addressing ambiguities through the use of black patches to mask ambiguous elements in the image, it is possible to improve initial performance significantly, aiming to maximize recall while maintaining high precision

However, we have demonstrated this only for a box size of 200x200. It would be valuable to verify whether this new training approach outperforms the previous approach based only on map-based automatic annotations for all relevant box sizes studied in Section 3.5. Therefore, we trained models with box sizes ranging from 50x50 to 250x250.

Figure 3.9 summarizes through PR curves the results obtained with M and M & S with black patches methods for the chosen box sizes.

Regardless of the box size used, these results consistently demonstrate that the new method outperforms M with highly promising gains. The curves of the two models only intersect at a specific point for the 100x100 size, showing a relatively

Table 3.2: *Average precision obtained after 300 epochs of training with different box sizes using images automatically labelled by* M *and* M & S *with black patches methods.*

Box size	50x50	100x100	150x150	200x200	250x250
M	21.2	39.2	42.5	43.2	38.9
M & S with black patches	51.5	58.3	52.6	50.7	47.9

Table 3.3: *MAE-x obtained after 300 epochs of training with different box sizes using images automatically labelled by* M *and* M & S *with black patches methods.*

Box size	50x50	100x100	150x150	200x200	250x250
M	4.01	5.84	8.90	10.84	11.76
M & S with black patches	1.89	3.68	5.37	8.90	8.38

minor difference in performance. Overall, the curves are consistently dominated by M & S with black patches.

This conclusion is further reinforced by the AP scores obtained for each curve, as summarized in Table 3.2. The improvement is particularly remarkable for the smaller box sizes, especially the 100x100 size, which leads to a model that emphasizes recall. However, for those aiming to optimize precision further, the optimal choice consistently appears to be 200x200.

Besides by comparing MAE-x obtained with M and M & S with black patches as summarized in Table 3.3, the annotation improvement particularly improves the reachable positioning accuracy for all box sizes. It is due to the annotation positioning improvement thanks to S method.

Examples of detections obtained using the M & S model with black patches during training are shown in Figure 3.10 using 100x100 and 200x200 box sizes for training. To prevent cluttered images, detections were filtered to retain only those with a score above 0.25. Examples such as those depicted in Figure 3.10a can explain the loss of recall for the 200x200 model. Conversely, examples shown in Figure 3.10b illustrate why the precision of the 100x100 model is generally lower. It detected two false positives compared to the other model. This can be attributed to better training of the 200x200 model in this case or the high proximity between the objects, making them difficult to detect with higher box sizes, notably due to the NMS. As illustrated in Figure 3.10c, the proximity makes detection difficult for the 200x200 model, though the 100x100 model also struggled to detect all the poles in this image. Finally, as depicted in Figure 3.10d, both models sometimes detect bollards, contrary to our intention. Nevertheless, the 200x200 model seems

less confident in identifying them as pole bases. In this thesis, we consistently use a box size of 200x200 in the next chapter

3.8 IMPACT OF ANNOTATION ERRORS ON BOX SIZE SELECTION

In the previous section, we demonstrated the feasibility of achieving pole base detectors using automatic annotations while addressing uncertainties in the annotation process. However, such annotations inevitably contain errors, as illustrated in Section 2.5.3. These errors typically fall into three categories: annotation positioning errors, missed pole bases (false negatives), and false annotations (false positives). They invariably affect the results, as evidenced by the performance of various models.

Additionally, we have observed that adjusting the box size during training for the same model can significantly impact the final detector's performance. It appears feasible to select box sizes to optimize either recall or precision for a given model. Depending on the box size, the results on the same images can vary, leading to false positives or missed pole bases. Despite our intention to avoid annotating objects like bollards, the detectors still occasionally identify them. These observations may also come from annotation errors, highlighting the importance of studying these different sources of error to better understand their impact.

3.8.1 *Reachable performance on manually annotated data*

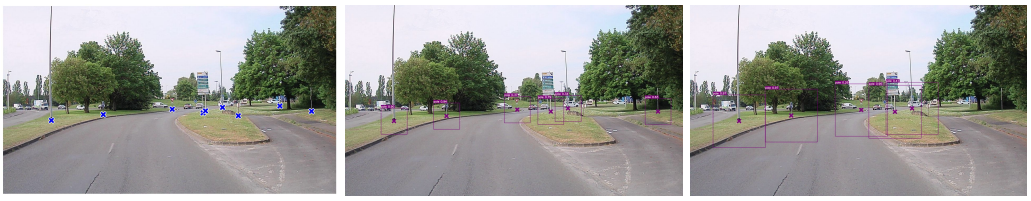
To assess the potential performance with a perfectly annotated training set, we split our manually annotated image set into two groups: a training set of 1927 images and a validation set of 903 images. It is important to note that even for humans, achieving perfect annotations is challenging, and errors may still be present, albeit in significantly fewer numbers.

We applied the same strategy as in Section 3.5 for the training and trained using multiple box sizes, ranging from 10x10 to 400x400. PR curves of all models are visible in Figure 3.11.

Optimal performance is achieved with a 25x25 box size. Sizes smaller than this are insufficient to reach this level of performance, indicating that the visual context is insufficient, or the box size does not properly cover the pole bases. For larger sizes, an increase in box size results in lower achievable recall for a given precision.

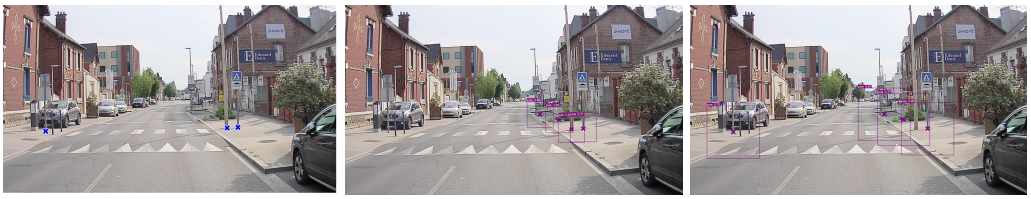
This behavior is similar to the one observed when using automatic annotations obtained from the HD map as shown in Figure 3.5, even if it is more pronounced here. The reduction in recall is logical because increasing the box size complicates the detection of close poles. Additionally, it results in more overlapping boxes, leading to potential suppression of accurate detections due to NMS.

However, the most notable difference from the study with automatic annotation is the achievable performance for a different optimal box size. With the map-based automatic annotations, we concluded that the optimal box size was likely



(a) *Example of pole missed by 200x200 model.*

(b) *Example of false positives by 100x100 model.*



(c) *Example of missed elements by both models. 200x200 model may have more difficulties due to higher box sizes.*

(d) *Example of bollards detection by both models. 200x200 model detects fewer bollards with lower scores.*

Figure 3.10: *Examples of detections on four manually annotated images using the M & S model with black patches. The left column shows the manual annotations. The middle and right columns display detections using a 100x100 box size and a 200x200 box size for training, respectively. To prevent cluttered images, detections were filtered to retain only those with a score above 0.25.*

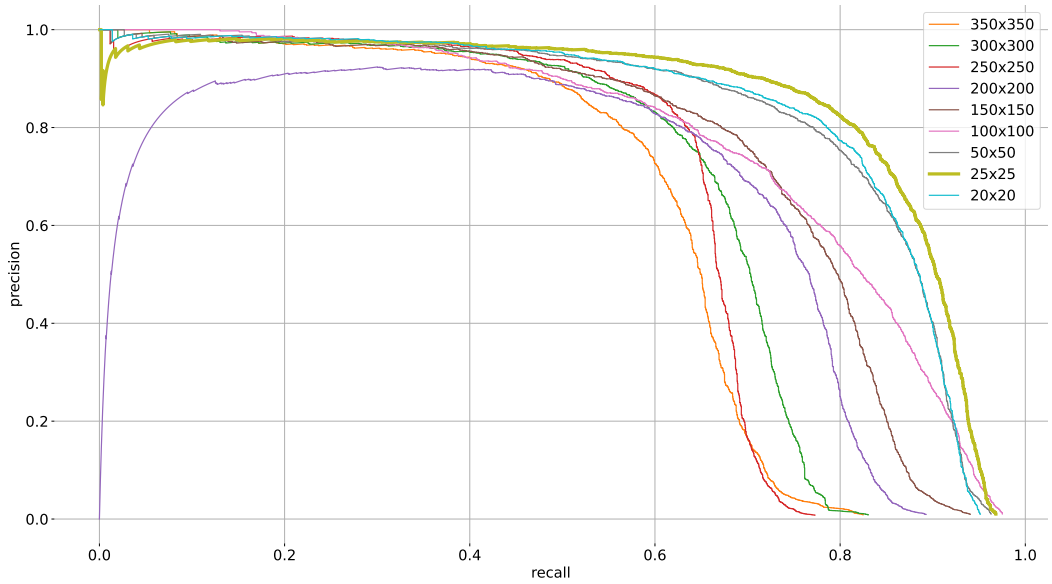


Figure 3.11: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images. The 25x25 box size offers the best performance as highlighted with its larger curve. Its PR curve consistently outperforms others across the majority of the precision/recall space.*

200x200. Here, it is clear that without automatic annotation errors, the optimal size is 25x25. The achievable performance with this optimal size is significantly higher than any previously observed network performance. Besides, for any box size, the performance exceeds those observed with the automatic annotations.

The most plausible explanation for such a difference lies in the potential errors that may occur during automatic annotation. Therefore, it is crucial to assess how these errors influence the network's performance.

3.8.2 *Simulated annotation errors on manually annotated images*

The errors in automatic annotation combine three potential sources: annotation positioning errors, missed pole bases (false negatives), and false annotations (false positives). Each of these sources may contribute differently to the observed performance differences. Therefore, it is crucial to analyze the influence of each error source.

To achieve this, we propose introducing errors into our manually annotated image set. To independently analyze each error source, a single type of error is added to the annotation set used for training. Thus, we propose three error simulation models inspired by those proposed by [Schubert et al., 2023].

To generate false positives (called spawns) to an image i, we select randomly annotations from other images. For the entire training set, we have a set of annotation A. We choose randomly a set of spawns A_{sp} to be added to the final training set such that $|A_{sp}| = \gamma_{sp} |A|$ where γ_{sp} is a spawn rate, i.e. the percentage of annotations chosen from the initial set to add spawns. It means that for

example, if $\gamma_{sp} = 0.5$, a third of the final annotation set corresponds to spawns. Each annotation of A_{sp} is assigned randomly to another image j. All the images in the datasets correspond to the same sensor and consequently have the same resolution, so all selected spawns can be added to any image.

The advantage of this method lies in the strategic placement of added spawns where the detector typically detects pole bases. These spawns are generated based on true annotations, ensuring they are positioned in areas of the image where the detector can anticipate finding a pole base.

To generate missed pole bases, called drops, to an image i, w can remove annotations for all images by selecting randomly $|A_d| = \gamma_d |A|$ where γ_d is a drop rate, i.e. the percentage of removed annotations from the initial set.

To add annotation positioning errors, we can transform each annotation $a_k^i = (u_k^i, v_k^i)$ from an image i by adding noises to obtain a new annotation $a_k^{i\,*} = \left(u_k^i + b_{u_k^i}, v_k^i + b_{v_k^i}\right)$ where $b_{u_k^i}, b_{v_k^i} \sim \mathcal{U}(-\epsilon, \epsilon)$ and ϵ the maximum error possible. For a new annotation $a_k^{i\,*}$, if it falls outside the image i, the annotation is then shifted to the edge of the image such that $0 < b_{u_k^i} < 1280, 0 < b_{v_k^i} < 720$.

It is crucial to note that the accumulation of these different error sources can lead to more pronounced degradation in performance compared to when only a single error source is present. However, this complexity makes analysis more challenging. Additionally, the simulated errors proposed here inherently differ from reality during automatic annotation of pole bases, thus the cumulative impact of simulated error sources would also differ. For instance, the positioning error model proposed differed from what we observed in x-axis errors during automatic annotation, as summarized in Figure 2.10.

The goal of this study is to identify a source that can have a greater impact than others and potentially provide clues on the most crucial correction to apply or subsequent analyses to do. The main objective is identifying error sources impacting performance regardless of the chosen box and those that may affect the box size selection.

To study the influence of each type of errors on the detection results, we propose analyzing the AP for multiple trained detectors with box sizes ranging from 20x20 to 350x350 after 300 epochs. This is done for different values of γ_d, γ_{sp}, and ϵ. The PR curves for each train with the tested parameter are provided in Appendix D, all leading to similar conclusions.

3.8.3 *Spawn influence*

Firstly, we trained multiple detectors for each box size with several values of γ_{sp} as visible in Figure 3.12, such that:

$$\gamma_{sp} \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\} \tag{3.7}$$

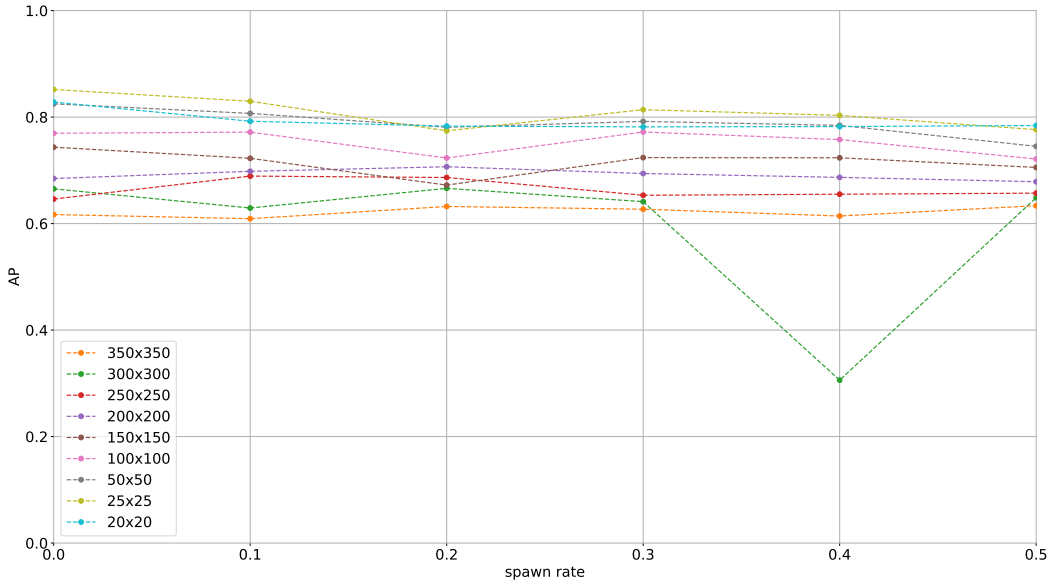


Figure 3.12: *Average Precision obtained after 300 epochs of training using manually annotated data with different box sizes and spawn rate. For a given spawn rate* γ_{sp}*,* γ_{sp} *percent of the annotation set is used to generate false positives.*

By analyzing the AP obtained for the different detectors, independently of the box size, the simulated added false positives seem to have a limited or no impact on the performance. The increase of the spawn rate seems to have the highest impact on the smallest boxes 20x20, 25x25, 50x50 and 100x100 with a small reduction of AP.

However, the performance difference seems to be quite limited or negligible, and in some cases, even positive, when increasing the spawn rate. This may be attributed to the spawn method's advantage that could have been a drawback. Adding spawns based on real annotations ensures that elements are inserted in areas of the image where the detector can anticipate finding a pole base. However, given the distribution of pole bases across all images, it is highly probable that spawns are added near already annotated pole bases. Therefore, as we increase the box size, their effect could potentially become entirely negligible.

3.8.4 *Drop influence*

Then, we trained multiple detectors for each box size with several values of γ_d as visible in Figure 3.13, such that:

$$\gamma_d \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\} \tag{3.8}$$

For any box size, except the 200x200 and 250x250 that seem to improve their performance when adding drops (Counter-intuitive and may be due to other reasons), if we neglect the small performance drops observed during different experiments, the AP seems to decrease with the drop rate, particularly for the smallest

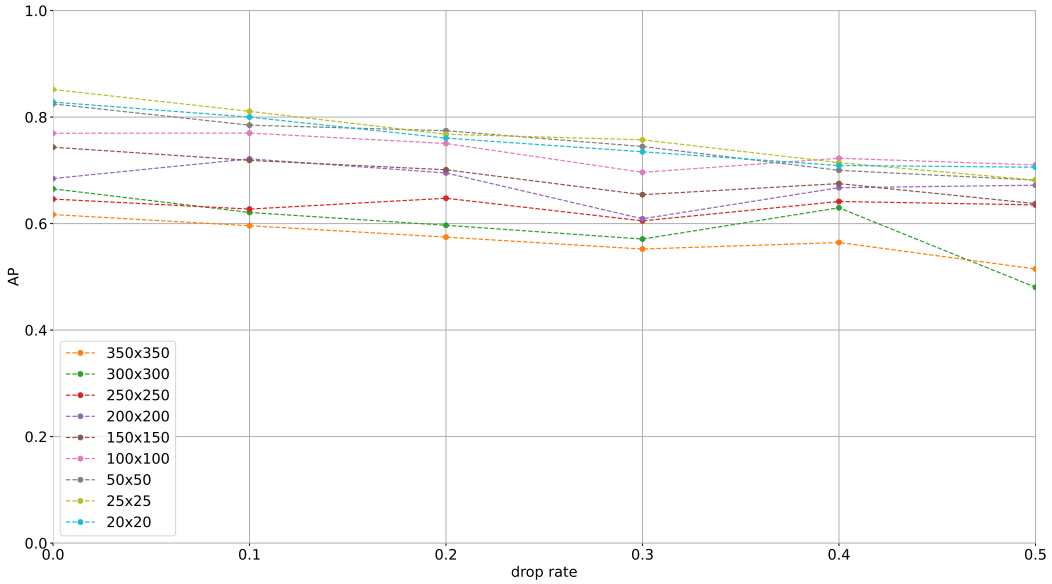


Figure 3.13: *Average Precision obtained after 300 epochs of training using manually annotated data with different box sizes and drop rate. For a given drop rate γ_d, γ_d percent of the annotation set is removed from the training set.*

box sizes. This behavior is much more noticeable than during the generation of spawns. Based on the error generation models used, it appears that the network is more sensitive to drops than to spawns.

3.8.5 *Noise influence*

Finally, we trained multiple detectors for each box size with different levels of positioning errors $\epsilon \in \{0, 2, 5, 7, 10, 12\}$ as visible in Figure 3.14.

For smaller box sizes up to 150x150, performance decreases significantly—the smaller the box size, the greater the performance loss. For larger box sizes, the added error appears to have minimal influence. This is logical because increasing ϵ results in shifting the boxes away from the pole bases. For smaller boxes, this shift eventually causes the pole bases to no longer be contained within the box. Additionally, as introduced in Section 3.4, detector evaluation relies on IoU between detections and boxes derived from annotations. Increasing the error increases the risk of shifting detections away from the pole bases, reducing the IoU with the boxes in the validation set, and thus decreasing the number of detections classified as true positives.

Following this study on the impact of errors on detector performance, it becomes evident that the network is particularly sensitive to errors in the positioning of annotations. However, this study is limited by its use of error models different from real errors committed by automatic methods.

For example, our latest detectors, trained using automatic annotations from maps and segmentation with black patches to manage ambiguities, significantly reduces both false positives (with annotation precision close to 90%) and false

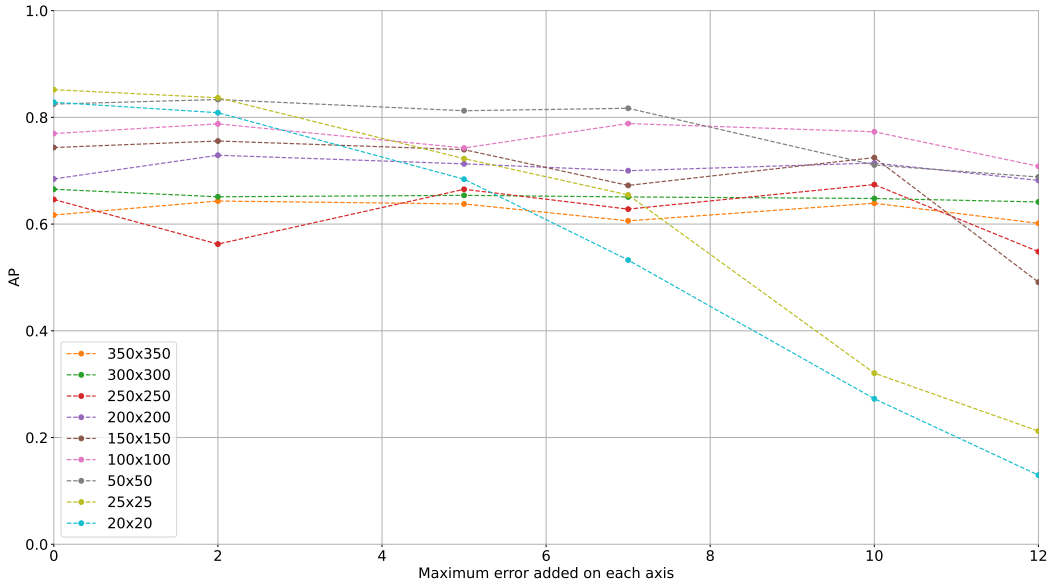


Figure 3.14: *Average Precision obtained after 300 epochs of training using manually annotated data with different box sizes and level of positioning errors.*

negatives (with recall close to 90%) if we assume that patches effectively masked ambiguous pole cases as intended. Positioning errors are also relatively small, mostly below 3px on the x-axis thanks to segmentation. Surprisingly, the optimal box size is not reduced, and performance remains limited compared to the performance reached in this study with the same box size.

Thus, the combination of these three error sources still appears problematic. However, focusing only on reducing the presence of drops and positioning errors can significantly improve results, given the outcomes of our study.

3.9 EVALUATION ON AUTOMATICALLY ANNOTATED DATA

A final interesting question here is whether it is absolutely necessary to manually annotate images to evaluate the detection performance. Since we can automatically annotate images, we can apply the same approaches to the validation set.

To experiment this, the same automatic annotation approach was used for both the training set and the validation set. The PR curves obtained with networks trained and evaluated on automatically annotated sets using the map-based method are shown in Figure 3.15.

Compared to the PR curves obtained on manually annotated sets as visible in Figure 3.5, the performance is highly overestimated in terms of precision and recall.

Nevertheless, we can observe a similar behavior: initially, increasing the box sizes led to a general improvement in performance, with optimal performance achieved for box sizes between 200x200 and 250x250, followed by a degradation in performance

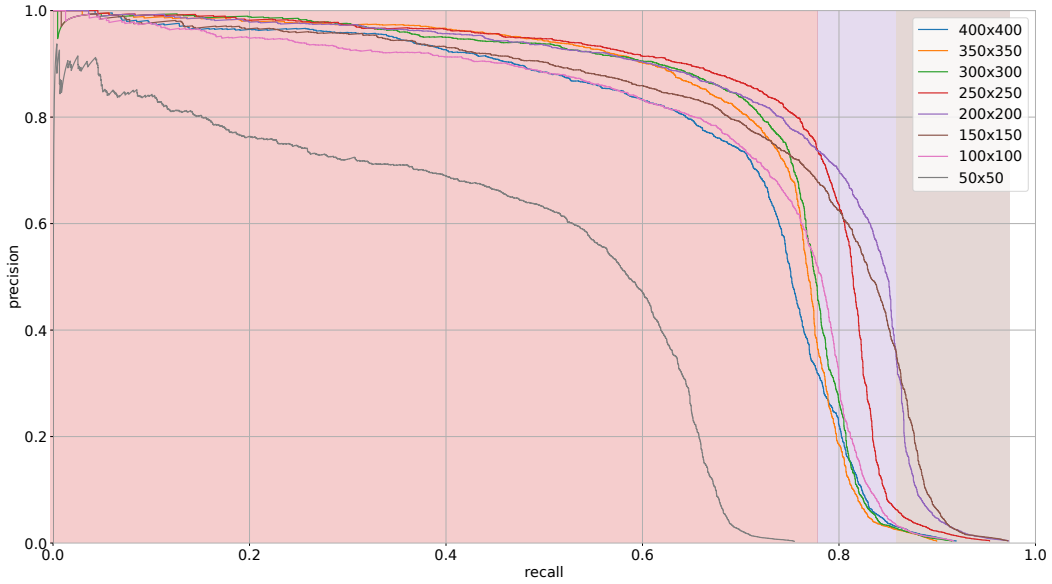


Figure 3.15: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using map-based automatic annotations for training and evaluation. The background color indicates the predominant curve. For simplicity, when the gap between two curves is too small, no background color change is applied.*

Using the automatic annotation approach with both the segmentation-based method and the map-based method, we evaluated the trained models previously obtained with the M & S method with added black patches on automatically annotated validation set using M & S without modifying the validation images.

The obtained PR curves are visible in Figure 3.16. Compared with the PR curves obtained previously on the manually annotated validation set in Figure 3.9, for the highest recall values (corresponding to the lowest score thresholds), the precision is significantly overestimated, and conversely, it is greatly underestimated for the lowest recall values (corresponding to the highest thresholds). Besides, in this case, it is more difficult to differentiate the performance reached with different box sizes, especially the unexpected difference between 100x100 and other box sizes observed in Figure 3.9.

It is therefore essential to maintain a small budget for manual annotation if one wishes to obtain the exact performance of the detector. However, if the objective is merely to find the optimal box size and then consistently use high confidence thresholds to ensure good precision, this becomes less imperative. Since the ultimate goal is to contribute to localization rather than achieving optimal detection performance, it is possible to partially reduce the use of manual annotation while being mindful of the limitations this implies on evaluating detection performance.

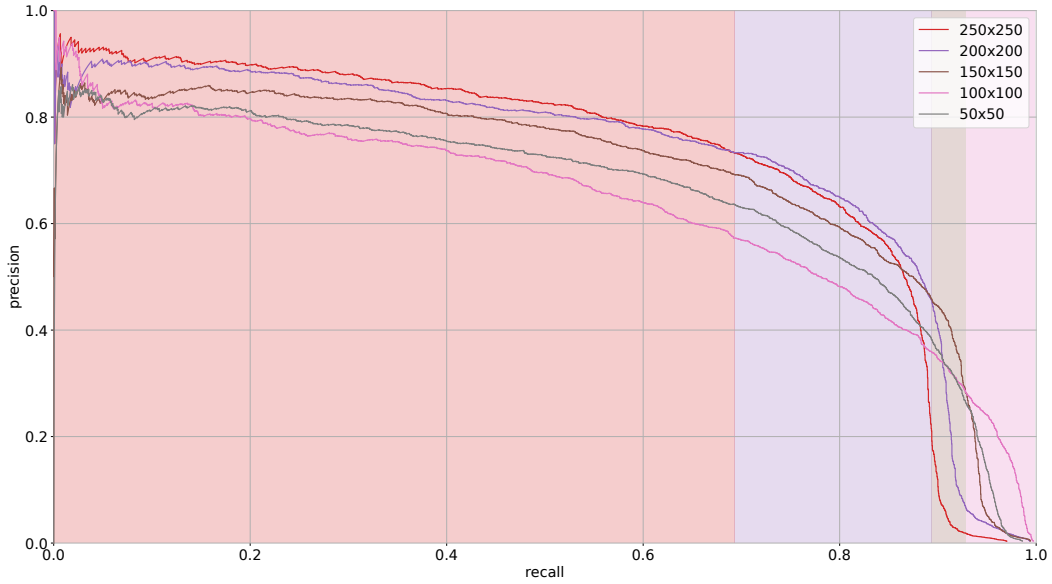


Figure 3.16: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using map-based and segmentation-based automatic annotations for training and evaluation. The training set is modified applying black patches to mask ambiguous cases. The background color indicates the predominant curve.*

3.10 CONCLUSION

In this chapter, we adapted well-known object detectors to detect pole bases using bounding boxes. It implied the determination of an optimal box size to provide enough visual context to accurately characterize the object.

To explore the possibility of training a pole base detector with automatically annotated images that align with the pole definition in our HD map, we trained YOLOv7 networks using annotations obtained from this map. This approach showed promising results with an optimal box size of 200x200, though performance was limited. We tried to enhance these results by integrating additional sources of automatic annotations. The various methods of combining these sources yielded different precision and recall behaviors.

Particularly, intersecting the map-based annotation set with the segmentation-based set improved precision, albeit at the expense of recall, likely due to some poles being left unannotated after the intersection. To mitigate this, we introduced a method to handle annotation uncertainties by masking parts of images with patches. It led to improved performance, surpassing the detector trained solely with map-based annotations. We found that this was consistent across all box sizes, with the optimal box size remaining unchanged, even after significant enhancements to the annotations, including better positioning accuracy. While the final results were impressive, they are still limited compared with those achieved with manual annotations, and the optimal box size also differed.

To determine the factors influencing performance and optimal box sizes, we analyzed the impact of various annotation errors, including false and missing

annotations, and positioning inaccuracies, on the training process. They significantly degraded performance compared with results with manual annotations, thereby affecting the choice of the optimal box size. However, the findings from this study differed from those observed with our automatic annotations, suggesting the need for further investigation.

At this stage, manual annotations were essential for validating the detectors' performance. However, to lower the costs of manual annotation, it is advantageous to consider using automatically annotated data for validation also. We demonstrated that using automatic annotations for validation led to similar conclusions regarding the optimal box sizes. However, the behaviors for each box size differed significantly from those previously observed, with either overestimating or underestimating overall performance.

Thus, fully replacing manual annotations with automatic ones for validation proves challenging. However, it is feasible to determine the optimal box size using automatic annotations. For localization purposes, exact detection performance might be less critical than identifying the optimal detector settings and score thresholds that balance high recall with strong precision or high precision with strong recall. Since we do not know if score thresholds apply on automatic validation or manual validation lead to comparable behaviors, additional investigations are needed to validate this approach.

Chapter 4

Enhancing Multi-Sensor Localization with Camera Pole Detections

Contents

4.1 Introduction

In the previous chapter, we developed pole detectors for images and studied methods to enhance their robustness, achieving high precision despite relatively limited recall. These detectors were trained using automatic annotations introduced

in Chapter 2, with map-based annotations derived from an HD map playing a central role. Given the importance of the HD map in our approach, the detector is designed not only to identify elements present in the HD map but also to detect similar features, specifically unmapped objects that should have been included. Performance was significantly improved by adding annotations from image segmentation and managing annotation uncertainties.

Our primary objective is to enhance vehicle localization, as outlined in Chapter 1. Although the information from HD maps has been used to train the image detectors, the ultimate goal is to leverage the georeferenced features stored in these maps to improve the accuracy of a multi-sensor localization system. Indeed, we position our work within the context of a multi-sensor system, as it is widely recognized by the scientific community that a single localization system cannot achieve the performance required for autonomous vehicles, especially in terms of robustness.

The question we address in this chapter is whether we can enhance the accuracy of a multi-sensor localization system by using our most precise pole detectors and quantify the performance gains based on real data. Additionally, we aim to determine whether the improvements made during detector training are crucial for achieving better localization.

This involves tackling the challenges specific to monocular camera-based detection within the context of localization. We begin by reviewing traditional methods commonly employed in pose estimation problems, then move on to identifying the unique aspects of our specific case. This process includes defining the various measurements used in our multi-sensor fusion method.

To evaluate the impact of the learning approach, we compare the localization improvements achieved by two models: one trained exclusively on map-based annotations, and the other using a combination of map-based and segmentation-based annotations with annotation uncertainty management.

Given our choice to employ a classical estimation method based on Kalman filtering, which may be highly sensitive to errors in the association between the HD map and detections, we focus on analyzing detectors with high precision levels of 95% or 90%. This approach helps us evaluate whether achieving the highest possible precision is essential or if a modest reduction in precision could still yield improvements in our case study.

There are various GNSS solutions with different levels of accuracy that can be integrated into a multi-sensor system. For instance, the SPP and PPP GNSS modes offer metric and decimetric accuracy, respectively. We propose to evaluate the potential accuracy improvement that can be achieved by incorporating additional camera-based pole observations to determine whether, within our context, even highly accurate solutions, such as those relying on PPP-RTK, can be further enhanced.

4.2 LANDMARK-BASED LOCALIZATION: A STATE-OF-THE-ART

4.2.1 *GNSS solutions for pose estimation*

In localization problems, various sources of information are usually merged to improve the final estimate. In outdoor environments, a GNSS receiver is frequently employed because it provides global positioning across the entire Earth without requiring specific infrastructure. In our context, GNSS not only aids in pose estimation but also plays a crucial role in linking visual detections to map elements.

GNSS positioning relies on multiple constellations such as GPS, Galileo, and Glonass, with each satellite transmitting diverse signals that can be used to estimate vehicle position. Each satellite regularly sends navigation data, which are binary-coded messages containing information on the satellite's ephemeris, clock bias, almanac, and other relevant data. For positioning estimation, a GNSS receiver uses ranging codes sent by the satellites, which are Pseudo-Random Noise sequences that enable the receiver to determine the travel time of the signal from the satellite to it by comparing them with sequences generated.

A GNSS system allows a receiver to compute for each ranging code i, a pseudo-distance, corresponding, as a first approximation, to the given model:

$$R_i = c\Delta t = c\left(t_{rcv} - t_{sat_i} + \delta t_{rcv}\right) \tag{4.1}$$

$$= \sqrt{(x_{rcv} - x_{sat_i})^2 + (y_{rcv} - y_{sat_i})^2 + (z_{rcv} - z_{sat_i})^2} + c\delta t_{rcv} \tag{4.2}$$

where $(x_{rcv}, y_{rcv}, z_{rcv})$ is the position of the receiver at the receiving time t_{rcv}, $(x_{sat_i}, y_{sat_i}, z_{sat_i})$ the position of the satellite sending the code i at the emission time t_{sat_i}, c the speed of light and δt_{rcv} the receiver clock bias which cannot be neglected. The coordinates are expressed in Earth-Centred, Earth-Fixed (ECEF) reference system.

There are thus four unknowns to determine. By using at least four codes, these unknowns can be estimated by solving the non-linear equation system.

However, multiple error sources affect the GNSS signals, impacting the distance measurements, as shown in Figure 4.1.

A pseudo-range measurement model can be then written as:

$$\rho_i = R_i - c\delta t_{sat_i} + cT_{gd} + d_{orb} + d_{trop} + d_{ion} + d_{rel} + \epsilon\left(\rho_i\right) \tag{4.3}$$

where δt_{sat_i} is the satellite clock error, T_{gd} is the satellite group delay, d_{orb} is the satellite orbit error, d_{trop} is due to the tropospheric delay, d_{ion} to the ionospheric delay, d_{rel} to the relativistic effects and $\epsilon_(\rho_i)$ is the receiver noise. Errors are consequently due to satellites and receivers components, atmospheric conditions and physical effects [Subirana et al., 2013].

[1] https://gssc.esa.int/navipedia/index.php/GNSS_Measurements_Modelling

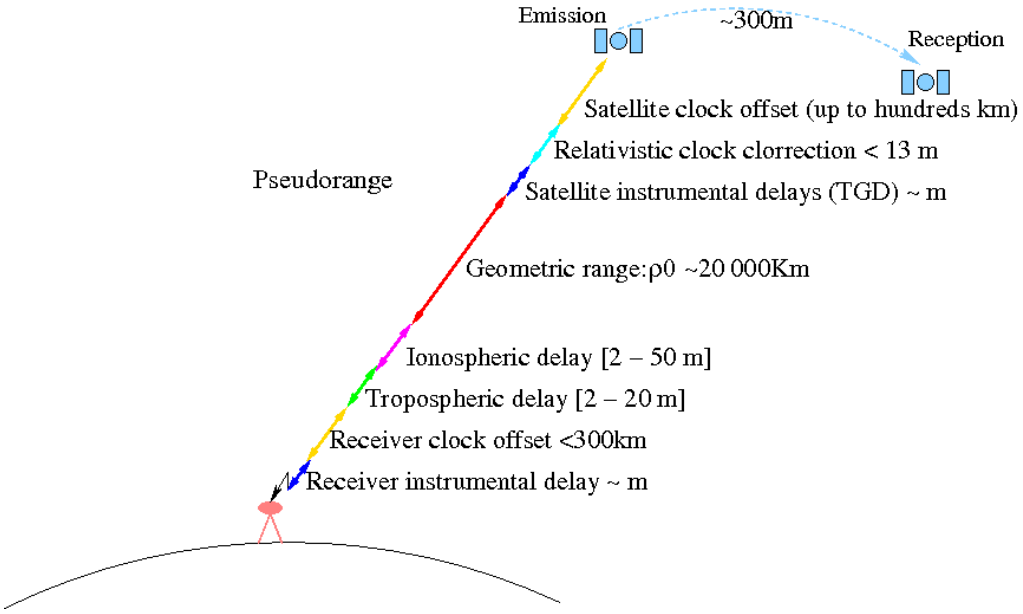


Figure 4.1: *GNSS error sources and magnitudes. From ESA Navipedia*[1].

The use of pseudo-ranges for positioning is called Single Point Positioning (SPP) [Subirana et al., 2013] and has a limited accuracy of few meters due to these error sources.

To reduce the effects of pseudo-ranges errors and to improve accuracy, additional measurements can be taken into account. Ground-Based Augmentation Systems (GBAS) and Satellite-Based Augmentation Systems (SBAS) can be employed. GBAS uses ground stations to provide localized code corrections, enhancing accuracy for specific areas like airport landings. SBAS, such as EGNOS in Europe, uses ground stations and geostationary satellites to broadcast corrections over broader regions. [Subirana et al., 2013]

Another effective method to significantly enhance positioning accuracy is Differential GNSS (DGNSS). This approach involves using (at least) a reference station located near the receiver (within a few kilometers). The reference station, with precisely known coordinates, computes corrections for pseudo-ranges and sends them to the receiver. This technique can achieve positioning accuracy within a meter.

Each ranging code is transmitted via a signal at a specific frequency, known as the carrier phase. Satellites can simultaneously send multiple ranging codes on different carrier phases, enabling more accurate positioning when using multi-frequency receivers. However, despite these advancements, the positioning accuracy remains constrained by the inherent accuracy of ranging codes.

To achieve higher positioning accuracy, advanced methods have been developed and use carrier phase measurements to refine positioning computed with code measurements. The carrier phase frequency, which is in the order of 1.2-1.5 GHz, is significantly higher than the code frequency of around 1 MHz. This

higher frequency enables much more accurate range measurements at the cost of additional unknowns to be estimated.

For each carrier phase Φ_i, we have:

$$\Phi_i = R_i - c\delta t_{sat_i} + cT_{gd} + d_{orb} + d_{trop} - d_{ion} + d_{rel} + d_{w_i} + \lambda_i N_i + \epsilon(\Phi_i) \tag{4.4}$$

where d_{w_i} is the phase wind-up, λ_i the wavelength and N_i the integer ambiguity. Since Φ_i carries code i, it is subject to similar sources of error. However, it also introduces two additional sources of error inherent to the sinusoidal signal measurement used here: phase wind-up and phase ambiguity. The phase wind-up d_{w_i}, a small effect caused by the orientations of satellite and receiver antennas. This effect, significant for achieving sub-decimeter precision, results in phase variations due to antenna rotation, which the receiver misinterprets as range variations. Phase wind-up can be corrected by accounting for relative rotations and the line of sight between the receiver and satellite [Wu et al., 1992].

Unlike code measurements, carrier phase measurements are ambiguous, requiring both the phase signal and the integer number of cycles N_i that the signal has done to travel the distance between the satellite and the receiver. Estimating this integer number is complex and necessitates advanced methods. To achieve high-precision positioning, several techniques manage error sources in a way similar to code measurements.

A prominent method is Real-Time Kinematic (RTK) positioning, similar in its principles to DGNSS. A reference station with known coordinates provides phase corrections to mobile receivers. The base station transmits its coordinates, corrections, and phase measurements. The mobile receiver uses this information to correct its measurements, significantly enhancing positioning accuracy. By applying differences of measurements, RTK can correct all error sources except for phase ambiguity. For this, the primary technique involves using double differences, comparing carrier phase measurements from two different satellites with those from the reference station for the same satellites [Carcanague et al., 2011]. This method helps to isolate the integer ambiguities and solve them. However, accurate ambiguity estimation assumes that all error sources (particularly ionospheric delay) are correctly mitigated. This is why RTK performance is typically reliable within 10-20 km of the reference station. This computation can be also carried out in post-processing to achieve greater accuracy for evaluation purposes or for further usage, as we have done with automatic annotation. In this case, this method is known as Post-Processed Kinematic (PPK).

RTK offers high positioning accuracy but has significant limitations. It necessitates a dense network of base stations and a constant connection for receiving corrections, which is often unavailable in many regions. To address this issue, Precise Point Positioning (PPP) has been introduced as an alternative. Unlike RTK, PPP does not rely on base station corrections for double differencing to estimate ambiguities and correct errors. Instead, it employs various methods to achieve high-accuracy positioning. PPP uses real-time corrections for satellite orbit and

clock errors from various services. To mitigate tropospheric delays, effective models are available, and while additional models can help with ionospheric delays, they are insufficient for delay correction. Minimizing ionospheric delay is crucial for accurate phase ambiguity estimation, as highlighted in RTK.

For that, multi-frequency receivers are essential in PPP. By combining measurements from different frequencies, it is possible to create ionosphere-free combinations [Héroux et al., 2001] that eliminate the first-order ionospheric delay. Then, for example, in a double-frequency context, Melbourne-Wubbena combinations can be used to remove remaining errors and estimate phase ambiguities [Subirana et al., 2013].

Various strategies can be used to achieve precise position estimates, but PPP's main limitations include the long convergence time required to resolve ambiguities, which can take tens of minutes to achieve sufficient accuracy. Additionally, reconvergence is needed after losing satellite signals. Incorporating more frequencies [Geng et al., 2020] allows reducing the convergence time. These challenges are significant obstacles for real-time vehicle positioning.

To address these limitations and offer a viable solution without solely relying again on RTK, researchers have developed a new method called PPP-RTK [Li et al., 2022]. This approach combines the benefits of both PPP and RTK while overcoming their respective drawbacks. To enhance the convergence time for ambiguity resolution, PPP-RTK uses precise corrections from a less dense network of ground stations, which is far less extensive than a typical RTK network. In fact, additionally to clock and orbit corrections, tropospheric and ionospheric corrections are estimated by ground stations and transmitted to the receiver. It allows developing new techniques reaching low convergence time below a minute and guaranteeing a high accuracy.

4.2.2 *Data association*

Traditional matching strategies typically use geometric characteristics, such as distances between measurements and map features, to identify accurate matches. While there are techniques that combine semantic and geometric information for data association, as proposed by [Pauls et al., 2020; Doherty et al., 2020; Denœux et al., 2014], our focus is on elements that lack semantic information. Therefore, we concentrate on geometric methods.

Some methods, like the one proposed by [Welte et al., 2020], aggregate multiple detections at different timestamps before association. This approach is particularly useful when the number of detections at a given time t is insufficient for reliable results, or when there is a high risk of errors. However, in this study, we focus on snapshot association methods that only manage the detections at each timestamp t of the current image. We made this choice because the association problem can take a long time to solve when many detections are made. Indeed, the process of associating detections with map elements can be inherently complex.

Several factors contribute to the complexity and potential errors in the association process. Firstly, false detections can be mistakenly associated to map elements, leading to errors in subsequent pose estimation. Furthermore, the detections themselves come with inherent uncertainties, complicating the association further. Even the map features do not always correspond to reality. Some elements may have been moved or even deleted. Others may have been added. In addition, there are mapping positioning errors. Then, to integrate detections from various sensors with map elements, it could be needed to express their positions in a common reference frame. This process employs reference frame transformations, as discussed in Section 2.3, and the uncertainty and accuracy of the pose influence the association results and may introduce errors.

We introduce in the following several association approaches, non-exhaustive, and we explain how they can be affected by various types of errors, as well as how they can help avoid certain errors in specific contexts.

Nearest Neighbor

The Nearest Neighbor is the simplest data association method. A specific association metric is selected, with the choice of metric space—whether it be the positional space or the sensor space—depending on the problem to be solved. The initial step involves defining a gating zone within which associations are considered feasible. Various distance measures can be employed in this process to ensure accurate association. To take into account uncertainties, it is possible to use the Mahalanobis distance between a map element m and a measurement y [Bar-Shalom, 1987]:

$$\mathcal{D}^{\text{mah}}_{m,y} = \sqrt{(m-y)^\top R^{-1}(m-y)^\top} \tag{4.5}$$

where R is the covariance matrix of measurement $\mathbf{y}$ supposed unbiased. In this case, we suppose there are no uncertainties on map features position.

As shown in Figure 4.2, different metric can provide different results. In this figure, the Euclidean and Mahalanobis gating zones are displayed. The association outcome using the Mahalanobis distance can differ significantly from that obtained with the Euclidean distance. Accurate estimation of the covariance matrices, represented by red ellipses, is crucial; any underestimation or overestimation can drastically alter the results.

As previously stated, a maximum value for the association metric has to be set to define the gating zone. When using Mahalanobis distance, this value can be set as a statistic test. In fact, if $y \sim \mathcal{N}(m, R)$, then $\mathcal{D}^{\text{mah}^2}_{m,y} \sim \chi^2_2$. Consequently, we can exclude wrong associations as much as possible while keeping a ratio $1-\alpha$ of good associations. The higher α, the higher the rejection rate of good associations. In practice, a risk $\alpha = 0.05$ is often arbitrarily chosen [Bar-Shalom, 1987]. With 2D features, an association is rejected if

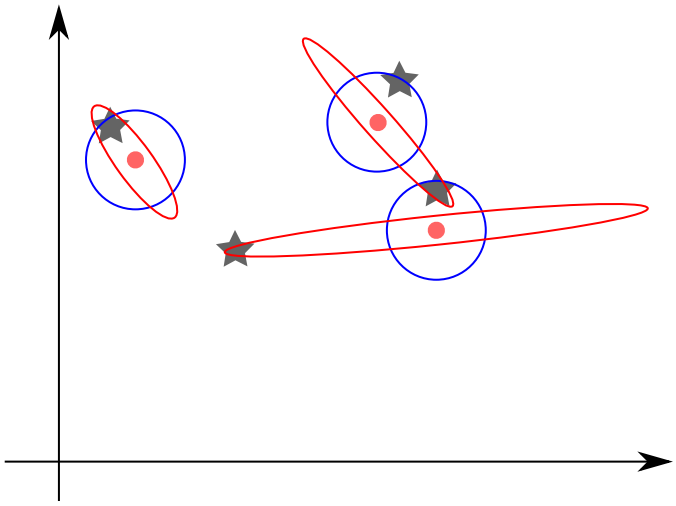


Figure 4.2: *Nearest Neighbor data association using Euclidean distance (gating zones in blue) and Mahalanobis distance (gating zones in red). Detections are circles and map features are stars.*

$$\mathcal{D}_{m,y}^{mah^2} > F_{\chi_2^2}^{-1}(1-\alpha) \approx 6 \tag{4.6}$$

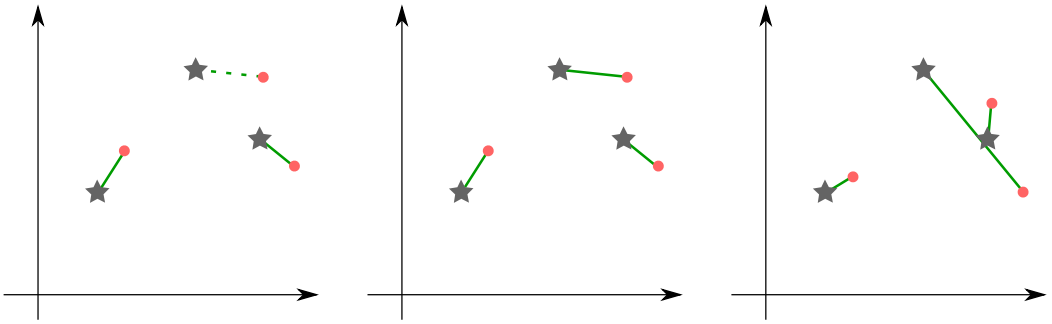
where $F_{\chi_2^2}^{-1}$ is the quantile function of the χ_2^2 distribution.

The Mahalanobis distance is particularly effective when there is high confidence in the accuracy of uncertainty estimates for both the detection and the pose used to transform within a common reference frame the map elements and the detections. However, if uncertainty is poorly estimated or the Gaussian hypotheses for association rejection threshold tuning are not fully satisfied, the Mahalanobis distance may lose its effectiveness. In such situations, it may not provide significant advantages over purely geometric methods for data association. Consequently, choosing a simpler metric, such as the Euclidean distance, might be equally valid.

If the sensor processing is designed to provide a single detection per feature, the primary drawback of using NN is that multiple detections can end up being associated with the same map feature, which is not ideal.

To prevent multiple measurements from being associated with the same map feature, the Unique Nearest Neighbor (UNN) method can be applied. In this approach, only the detection with the smallest distance is retained, while the others are excluded from the data association process.

However, valuable associations can be lost. As illustrated in Figure 4.3a, if a detection error places a detected feature close to two different map features, a valid detection might be discarded and left unmatched. Another approach (that we call Alternative NN) then consists in associating detections iteratively in ascending order of distance. When a map feature is associated with a detection, it is removed from map feature candidates for other detections. This allows associating the detection that was discarded by UNN as visible in Figure 4.3b but this can also create erroneous associations as shown in Figure 4.3c.



(a) *UNN* ***(b)*** *Alternative NN* ***(c)*** *Alternative NN*

Figure 4.3: *Example of different UNN and alternative NN association results. Stars represent map features, and circles represent detections. Green lines indicate the associations made. The gating zones are not shown to avoid cluttering the figure. In (a), the dotted line represents a true association that was not established due to the UNN strategy. In (b), a correct association is shown. In (c), an incorrect association is illustrated.*

Hungarian Method

Global approaches can also be applied to solve data association. Munkres algorithm [Kuhn, 1955] allows association of map features with single detections, by not necessarily associating a measurement with the closest map feature.

Let Y be a set of detections, M the set of map features, and Z a set of couples of detections and map features corresponding to potential associations. The Hungarian method solves:

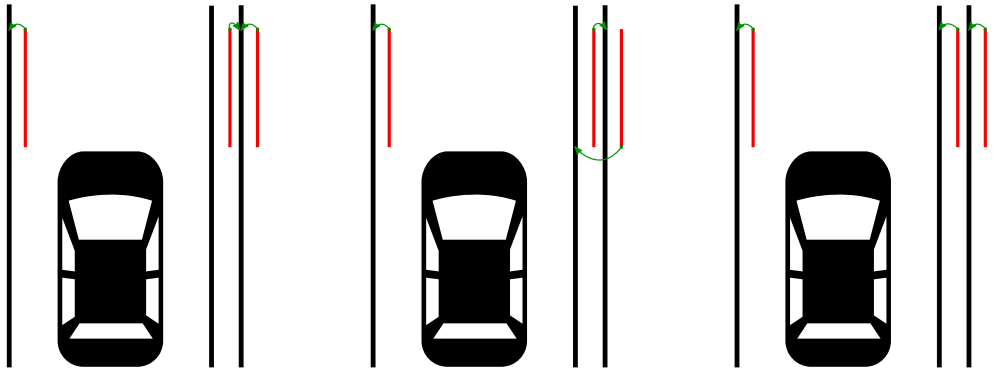
$$Z = \underset{Z \subset Y \times M}{\arg\min} \left(\sum_{\{y,m\} \in Z} \mathcal{D}_{y,m} \right) \tag{4.7}$$

where $\mathcal{D}$ is a distance function.

The data association problem is seen as an assignment problem where distances are assignment costs and detections are associated to any feature if it decreases the global assignment cost.

To manage wrong detections and avoid them being associated to a map feature, two options are available compared to NN strategies. During the assignment problem, when calculating distances, if some exceed the defined threshold for gating, we can replace them with an infinite cost, forcing non-association and effectively managing the gating zone. Alternatively, the assignment problem can be solved without considering the gating zones, and then all associations with distances exceeding the gating threshold are rejected. These two approaches can lead to different outcomes. In the remainder of this thesis, when the Hungarian method is applied, we use the second strategy.

When focusing on 2D features, it could lead to similar issues as those seen with NN alternative in Figure 4.3c, but solve the association problem with a more coherent global approach than this algorithm.



(a) *NN* ***(b)*** *Alternative NN* ***(c)*** *Hungarian method*

Figure 4.4: *Data association strategies for lane markings map-matching. Detections in red are misaligned due to a localization error.*

The Hungarian method is much more efficient than all the methods seen so far particularly when trying to associate 1D elements. Consider the example of lane markings, which provides a particularly clear visual illustration as shown in Figure 4.4. We can see with Figures 4.4a and 4.4b that applying UNN or Alternative NN led to a loss of information or erroneous associations respectively. Munkres algorithm makes the best use of all the information of perception in such a case as shown in Figure 4.4c.

None of the previously discussed methods account for maintaining consistency between the relative distances of detections and the relative distances of map features. Observations and landmarks are treated independently of each other, without consideration for the spatial relationships among them. This means that the spacing between detections and their corresponding map features can differ, leading to significant 'deformations', particularly in 2D scenarios.

Moreover, all the methods introduced here require a good prior estimation of the vehicle pose to bring the detections and the map features in the same reference frame for data association.

Graph-based and pattern-based methods

Methods guaranteeing consistency of map geometric structure and relaxing the constraint on initial pose estimation exist. It is particularly the case of Combined Constrained Data Association (CCDA) [Bailey, 2002].

It is a data association strategy using graph cliques to find consistent matches. A graph is built where each node is a pair (y_i, m_j) of an association candidate. The nodes are connected to each other if the Euclidean distance between the measurements of the candidates is the same as the distance between associated map features. Graph edges are added between pairs (y_i, m_j) and (y_k, m_l) if $|\mathcal{D}_{y_i,y_k} - \mathcal{D}_{m_j,m_l}| < \tau$, with τ a threshold. The biggest clique, i.e the biggest complete subgraph, is then maximum consistent matches. An example where all association candidates are used in the graph is shown in Figure 4.5. In this figure, Euclidean distances are considered.

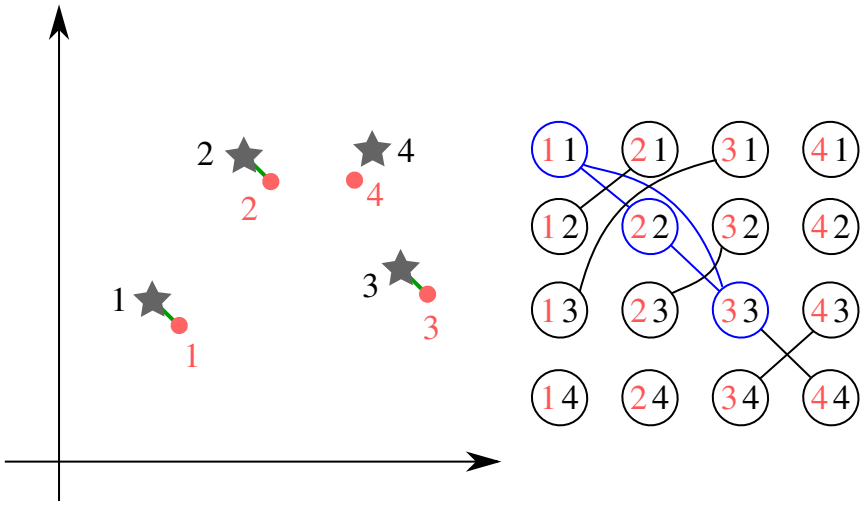


Figure 4.5: *Combined Constraint Data Association example. Green links are obtained associations. Nodes are association candidates. An edge is drawn between 2 nodes if the distance between measurements is similar to the distance between map features, with a limited tolerance. The associations obtained correspond to the biggest clique highlighted in blue. The measurement 4 is not associated because it is not consistent with the map geometry.*

In this example, the problem is relatively straightforward and could be addressed with alternative pattern matching methods. Typically, in Section 1.2, we introduced ICP in a dense maps context to align point clouds. It consists in minimizing iteratively the distance between the two sets of points to determine the transformation between the sets. By removing the star n°4 in Figure 4.5, we can see that the two sets of points can be aligned effectively using ICP. In [Li et al., 2021], ICP has been used to match detected pole-like features with their equivalent in a vector map. It was pointed out in the litterature that ICP has several drawbacks as slow convergence, sensitivity to wrong detections and missing detections.

CCDA is unfortunately NP-hard, meaning its complexity increases exponentially with the number of nodes, making it impractical for scenarios with large numbers of detections or map features. Despite this, it remains valuable in cases where the prior pose estimate is highly uncertain. By constructing a graph that incorporates all available map information in the vicinity of the pose, CCDA effectively manages data association under challenging conditions, though this comes with the trade-off of potentially high computational time. To reduce the number of nodes, information on map features and measurements can be used as the geometric proximity between measurements and map features (an initial pose is needed) or proximity in terms of semantic information.

Nevertheless, it still faces limitations. As introduced in Section 1.4.2, in ambiguous geometric configurations, which can usually occur very regularly in an entire map, multiple association solutions are possible. Using a prior pose for data

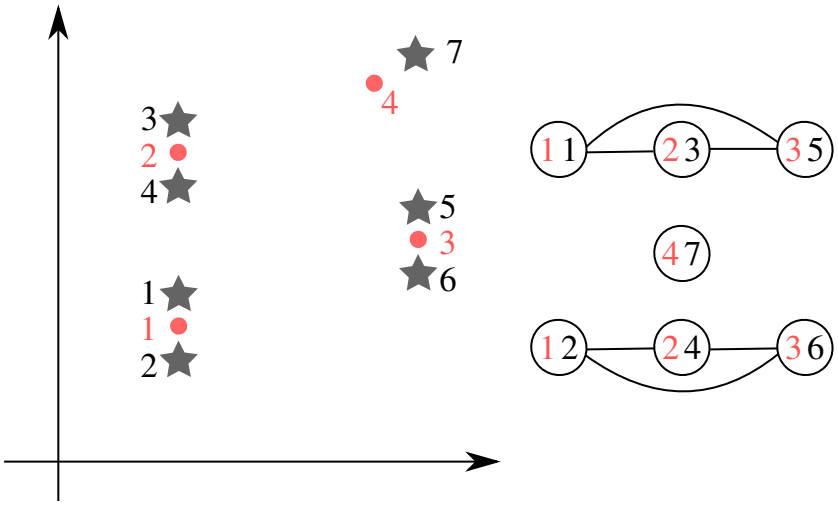


Figure 4.6: *Combined Constraint Data Association with map ambiguities.*

association can help tackle this issue, but it is still possible to observe such ambiguous configurations in a local environment as shown in Figure 4.6 where two association results are possible even with methods like CCDA or ICP.

Other methods based on pattern matching can be used to get relative distances consistency. In [Hofstetter et al., 2021], hashing methods are used. In a first offline step, patterns called constellations are extracted from a map. Each constellation is coded by hash values in a table based on local geometry. Then in an online step, from detection of multiple features, constellations are built. Correspondences between map and detection are found based on computed hashes. Thanks to this method, it is possible to localize itself relatively to the map and globally without prior knowledge. Nevertheless, an offline learning phase is needed and the number of constellations can be particularly high. Besides, the robustness against unmapped features is not always guaranteed. As CCDA, such a method is sensitive to geometric ambiguities in the map data. That is why, the authors in [Hofstetter et al., 2019] focuses on detecting map ambiguities during an offline phase using this technique. Then, it helps not use the ambiguous patterns during an online data association phase. In [Cao et al., 2020], a matching between grid maps is applied to solve the data association through pattern matching. In [Steinke et al., 2021], map features are encoded by their own characteristics and their fingerprint, relative distances and angles between the features and its surroundings in a given radius. When features are detected, their fingerprints are built and the data association process looks for similar fingerprints in the map database.

4.2.3 *Data fusion strategies and visual observation for pose estimation*

To fuse information from multiple data sources and particularly in a landmark-based localization, multiple filtering techniques exist for properly fusing data, including GNSS data, IMUs, and detected objects. [Konrad et al., 2018] introduced multiple filtering approaches used for GNSS/INS fusion and can be used with all

data sources. The Kalman Filter (KF) and its extension, the Extended Kalman Filter (EKF) for non-linear dynamic systems are widely used. They allow estimating the state of a dynamic system using a series of noisy measurements, continually updating the estimate of the system state as new measurements become available. The filters combine information from the system's dynamics model and measurements to produce a more accurate estimate of the true state than would be possible using either source of information alone. The estimate is a vector of random variables modeled by normal distributions, for which we aim to characterize the mean and variance. To achieve this, the dynamic model and the measurements also assume white and centered Gaussian noises. Other filter approaches are possible such as the Particle Filter where the estimate is represented by a set of particles, each particle progressing by taking into account model and measurements and random noises modeled by any distributions.

When fusing GNSS and INS data, two strategies can be applied: loosely-coupled (LC) or tightly-coupled (TC). In an LC system, the GNSS and INS components operate independently. GNSS receivers provide a position estimate as output which is fused with data from INS. This approach offers flexibility and allows for easier system upgrades or modifications. In a TC system, the GNSS and INS components are tightly integrated at a lower level, with the raw GNSS measurements directly fused within the navigation algorithm. TC integration often results in superior localization performance in terms of accuracy, reliability and robustness. The main advantage of the TC integration is the possibility to update the localization estimate also in scenarios with few GNSS measurements (in particular when there are less than 4 satellites in view) and can decrease the influence of the varying satellite constellation. These two types of integrations can also be extended to add perception data into the fusion process. In this research, we have a preference for the TC scheme for similar reasons as with GNSS integration to be able, in particular, to make use of a few detections of pole bases, as these may be sparse. This will be further validated in the experimental section of the chapter, where we will observe that, in many cases, only a limited number of map-matched features are available.

Possible improvements in positioning can be in terms of 2D positioning, heading, cross-track accuracy or along-track accuracy. They are inherently limited by the capabilities and limitations of the sensor used: some sensors may excel in providing accurate lateral (cross-track) positioning, but they may struggle with inaccurate longitudinal (along-track) measurements.

In this chapter, we focus on the use of cameras. Indeed, cameras can quite easily detect lane markings and curbs and, when used in conjunction with HD maps, their measurements can significantly enhance cross-track accuracy to achieve lane-level positioning [Frisch et al., 2018; Al Hage et al., 2019]. However, due to the lack of inherent 3D information in camera data, improving along-track accuracy is more challenging.

Beyond lane markings, other road infrastructure elements, such as traffic signs, can serve as additional information sources. Monocular cameras typically provide

Table 4.1: *Detection performance and score threshold to be applied to* M *and* M&S *models to reach 90% and 95% precision*

precision (%)	M		M & S	
	recall (%)	threshold	recall (%)	threshold
95	4.3	0.956	30.4	0.8867
90	33.3	0.22	38.5	0.429

only angular information, resulting in a bearing-only localization problem [Jensfelt et al., 2006; Lemaire et al., 2007]. This type of bearing information has been applied across various domains, such as aviation and submarine navigation, as well as in vehicle localization. For example, [Hoshino et al., 2016] proposed to use an omni-camera and four distinguishable landmarks optimally placed to localize an automated agricultural vehicle. Bearing-only SLAM has also been widely studied [Bekris et al., 2006].

4.3 POLE-AIDED LOCALIZATION USING MULTI-CAMERA SYSTEM: PROBLEM STATEMENT

Pole-based localization is very challenging for data association due to the non-discernability of road features. Incorrect associations of pole detections with vector map features can therefore lead to poor localization. Moreover, relying solely on bearing information makes it difficult to accurately estimate a vehicle's pose since angular measurements are made relatively to the heading of the vehicle.

4.3.1 *Proposed detectors*

Through the implementation of various automatic annotation techniques and the management of annotation uncertainties, we developed two models able to detect mapped poles: M trained only using map data and M & S with enhanced uncertainty handling, which significantly boosts performance, especially in terms of precision.

As this performance boost is in terms of object detection, one can wonder whether it is necessary to refine the annotation approach to such an extent to achieve better performance in terms of vehicle localization. We therefore propose to compare the localization performance using both models. To minimize incorrect associations between detections and mapped poles, we set confidence score thresholds for both models at the same high level of precision, ensuring a fair comparison. As summarized in Table 4.1, we selected thresholds for both models to guarantee precisions of 95% and 90%.

Lowering the detection precision allows detecting more mapped features, but it introduces false positives into the localization process. Due to our detection

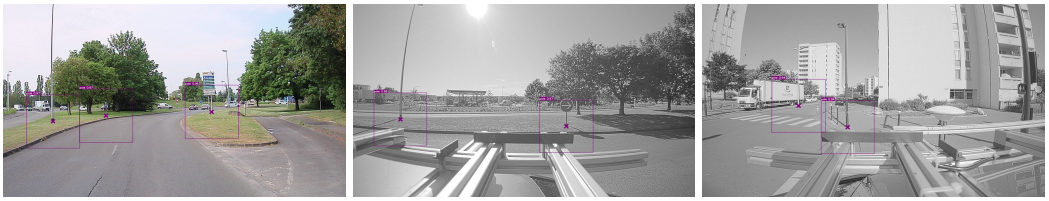


Figure 4.7: *Examples of detections obtained from YOLOv7-based pole detectors on RGB (front) and grayscale images (lateral). Each bounding box is displayed with its detection score and its center corresponding to a potential detected pole base is highlighted with a cross.*

improvements for these levels of precision, the recall was higher for M&S than for M, especially at 95% precision (7 times higher).

In chapter 3, we trained detectors for a color camera looking at the front of the vehicle. However, in a road environment, poles are present all around the vehicle. It may be beneficial to explore the impact of detecting poles in other directions to improve localization. To this end, the vehicle was equipped with two wide-angle grayscale cameras, one oriented towards each side.

Since these cameras are completely different from the previously used color camera, we cannot apply M and M & S previously trained. Instead, by using the same automatic annotation strategies, we developed two new M and M & S models customized for these cameras. Figure 4.7 shows examples of detections from each camera type, using M & S with a confidence score threshold of 0.8867, which corresponds to a 95% precision for the front-facing camera.

Since we lacked manual annotations for the grayscale cameras, we were unable to directly evaluate the performance of the models tailored for these cameras. Therefore, during the localization process, we applied the same confidence score thresholds used for the color camera, assuming that the precision were comparably high.

Due to the dependency on bounding boxes for detecting pole bases, there were instances where the centers of the boxes of correctly detected pole bases, in the sense of Chapter 3, do not correspond to the actual pole base. This was observed for detections near the image borders, where the box was cropped, as shown in Figure 4.8. To avoid creating inaccurate measurements that could have significantly affected pose estimation performance in next steps, we started by rejecting all boxes with centers within 100px of the left or right edges of the image, given our chosen 200x200 box size. The top and bottom edges were disregarded, as horizontal errors were considered more critical than vertical ones.

4.3.2 *Pose estimation*

We built a localization solution using an Extended Kalman filter. The system uses GNSS, wheel speed sensors, a gyro, three cameras for pole detection, and a vector

Figure 4.8: *Examples of wrongly detected pole base due to cropped bounding box near image edge*

HD map. The GNSS is loosely coupled, while the camera measurements are tightly integrated

The operating equations of an extended Kalman filter are not recalled in this manuscript. We refer the reader to the extensive literature on this subject. We simply point out that all the measurements are processed during the filter update phases and asynchronously as soon as they are available. Besides, the pose estimate is provided at the highest frequency possible corresponding to the highest frequency of the available sensors: 50Hz. The results presented below were obtained in post-processing, with the latencies of the various sensors disregarded for simplicity.

At a time k, the state vector $\mathbf{x}_k$ is expressed as follows:

$$\mathbf{x}_k = \left[x_{B,k}, y_{B,k}, \theta_{B,k}, v_k, \dot{\theta}_k\right]^\top \tag{4.8}$$

The component $\mathbf{q}_k = (x_{B,k}, y_{B,k}, \theta_{B,k})$ is the vehicle pose, *i.e.*, position and heading, defined at the center of the vehicle rear axle, corresponding to the center of the body frame B. The components v_k and $\dot{\theta}_k$ correspond to the longitudinal speed and the yaw rate, respectively.

At this stage, our goal is to determine whether pole base detections using cameras can positively contribute to the pose estimation process, even when using simple filters. Consequently, no fault detection and exclusion phase has been added, and no other, more robust filters have been examined. As a result, this filter is inevitably sensitive to errors, particularly association errors that are likely to occur.

4.3.3 *Bearing measurements on poles*

From an image, as explained by Eq. (3.2) and highlighted in Figure 4.7, bounding boxes are characterized by their dimensions, centers and confidence scores. Here, we assume that centers should correspond to pole bases candidates.

Consequently at a timestamp k, for a given camera C (which can be one of the three chosen cameras), we can deduce a set of detections in its image frame I after applying a score threshold $\mathcal{T}$:

$$ {}^{I}\mathbf{Y}_{k}^{C} = \left\{ {}^{I}\mathbf{y}_{k,i}^{C} = (u_{k,i}, v_{k,i}) \middle| \, i = 1, \dots \right\}, \tag{4.9} $$

where each detection ${}^{I}\mathbf{y}_{k,i}^{C}$ is the pixel coordinates u, v of a pole base candidate expressed in the image frame I of the camera.

4.3.4 *Data association of the bearings with the map features*

To create an observation model from camera measurements that the filter can use to correct its estimation, it is essential first to associate the camera detections with the mapped elements.

For this, the detections from the camera need to be expressed in a common space with respect to the map features. As highlighted by Eq. (4.9), camera measurements are only 2D data in an image, and no depth information is provided. However, as illustrated in Figure 2.2 with the camera model, projecting image points into a frame external to the camera requires this crucial depth information.

Consequently, we can only project the map features into the image frame, as done during automatic annotation using map information and a reference pose as explained in 2.3, to perform the data association in frame I with 2D elements. On an image with detected poles by the pole base detector and using a reference pose, this leads to the results highlighted in Figure 4.9.

From this projection, it is evident that the ability to associate detected features is limited to the u-coordinate in the image due to the map's lack of height information, resulting in a 1D data association problem. In fact, to enable image association using 2D coordinates, we might have attempted to correct the height of the map projection within the image, as done in Section 2.3 for map-based automatic annotation with a lidar and a reference pose of the vehicle. However, when working with a less accurate pose estimate, such as the current filter estimate, inaccuracies can introduce significant errors in height correction, adversely affecting the data association. Additionally, relying solely on lidar is not ideal and restricts the flexibility and usability of the method.

With precise detections and the use of a reference pose, associations can be accurately resolved in the image using the u-coordinate, but, in practice, projecting map features into the image frame at a given timestamp k involves using a pose estimate. Most of the time, this estimate is provided by the filter itself. It is inevitably affected by errors, which impacts the projection of the mapped poles.

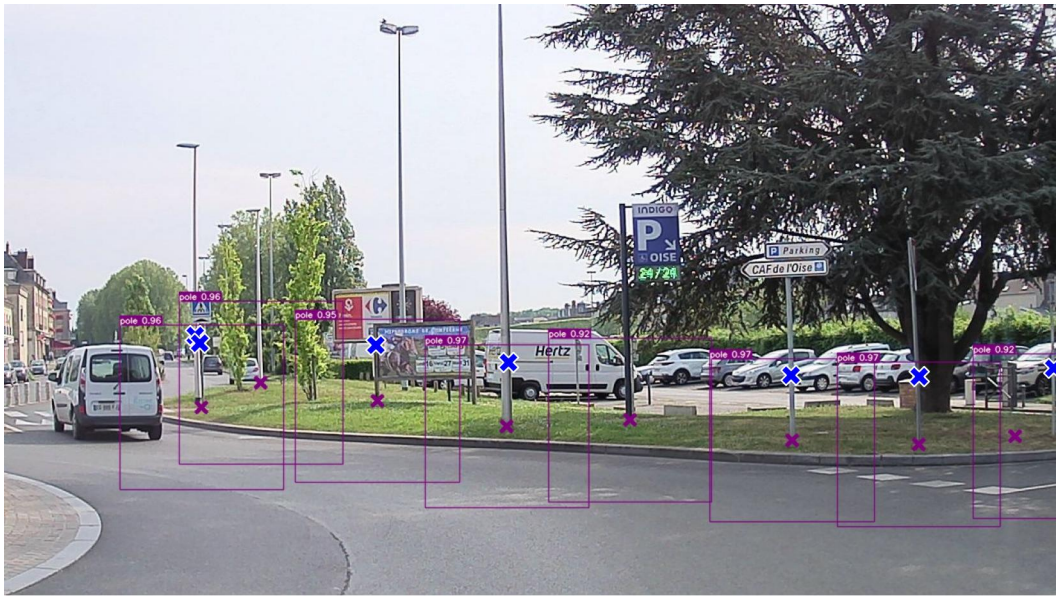


Figure 4.9: *Map projection in image frame using a reference pose in comparison with detection results. Map projections are highlighted with blue crosses and detections are visible in purple. Due to 2D assumption, the map is wrongly projected and the v-coordinate is unusable for data association whereas the u-coordinate is highly precise in this case. On the right side of the image, the issue of cropped boxes is clearly visible, as there is a significant discrepancy in the u-coordinate between the detection and its corresponding map feature.*

Consequently, some poles from the map visible in the image may not be properly projected or might not even appear in the image at all. In fact the true field of view of the camera would not necessarily correspond to the predicted field of view using the filter estimate. Besides in the image, poles close to the camera or behind it are not projected at all and would not be associated to any detection, even if they correspond to the detected objects. Therefore, instead of working within the image, it may be more effective to do the data association directly in the C frame, thereby avoiding image limitations.

Generally, the projection is influenced by the accuracy of the pose estimate, the sensor calibration both intrinsic and extrinsic parameters. These different problems can make the association problem particularly difficult in certain situations.

The results of projecting both detection and map features in frame C with a reference pose to solve the data association are shown in Figure 4.10.

Given the camera's intrinsic calibration parameters, the image detection set ${}^{I}\mathbf{Y}_{k}^{C}$ is transformed into a set of bearing angles expressed in the camera frame C:

$$ {}^{C}\mathbf{Y}_{k}^{\alpha} = \left\{ {}^{C}\mathbf{y}_{k,i}^{\alpha} = \alpha_{k,i} \in [-\pi;\pi) \middle| \, i = 1,\dots \right\} \tag{4.10} $$

where $\alpha_{k,i}$ corresponds to the angle of the i-th detection with respect to the direction pointed by the camera. The angle is deduced from Eq. (2.4) after applying distortion correction as:

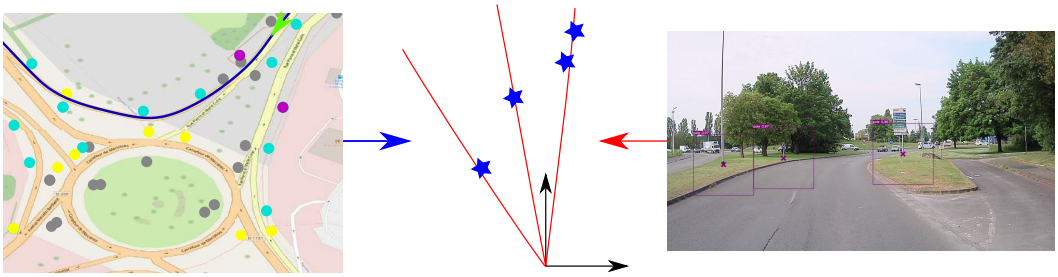


Figure 4.10: *Example of detections and georeferenced features projection in a common working frame for data association using a reference pose. Projected features from the HD map in the camera frame are visible in blue. The bearings in the camera frame obtained from the detections are visible in red. The position of the vehicle is indicated in green on the HD map.*

$$\alpha_{k,i} = \operatorname{atan}\left(\frac{u_{k,i} - c_x}{f_x}\right) \tag{4.11}$$

To do the association of the detected bearings, the map features are projected into the camera frame C:

$${}^{C}\mathcal{M}_k^{\alpha} = \left\{ {}^{C}\mathbf{m}_{k,j}^{\alpha} = \alpha_{k,j} \in [-\pi;\pi) \middle| \, j = 1,\ldots \right\} \tag{4.12}$$

To express the map features in the frame C, they are transformed as done in Section 2.3 using the estimated pose and the extrinsic calibration parameters. The bearing angles are deduced from the obtained coordinates. To prevent unlikely associations with features that are too distant from the vehicle, we restrict the map features to a specified radius around the pose estimate used.

In the camera frame, we choose to use the following distance between a camera measurement ${}^{C}\mathbf{y}_{k,i}^{\alpha}$ and a projected map feature ${}^{C}\mathbf{m}_{k,j}^{\alpha}$ to solve the association problem:

$$\mathcal{D}_{k,i,j}^{C} = \left({}^{C}\mathbf{y}_{k,i}^{\alpha} - {}^{C}\mathbf{m}_{k,j}^{\alpha}\right)^2 \tag{4.13}$$

The Hungarian method is used to associate ${}^{C}\mathbf{Y}_k^{\alpha}$ with ${}^{C}\mathcal{M}_k^{\alpha}$.

We selected this method because, UNN empirically delivered comparable results. We also considered that the Hungarian algorithm might provide additional robustness in scenarios with numerous detections. When the number of detections is low, the Hungarian algorithm and UNN produce similar results. To observe significant differences compared with Hungarian method, exploring more complex association approaches will be needed.

Since the bearing errors will be modelled in our experiments with the same distribution, a Mahalanobis distance would give similar results.

Once the data association step is done, the observations from the cameras are injected into the update stages of the localization filter. For a pair $\left({}^{C}\mathbf{y}_{k,i}^{\alpha}, {}^{C}\mathbf{m}_{k,j}^{\alpha}\right)$ of a detected pole base and its corresponding map feature, the observation model is as follows:

$$ {}^{C}\mathbf{y}^{\alpha}_{k,i} = {}^{C}\mathbf{m}^{\alpha}_{k,j} + \boldsymbol{\beta}^{\alpha}_{k,i} \tag{4.14} $$

where ${}^{C}\mathbf{m}^{\alpha}_{k,j}$ corresponds to the predicted bearing angle (and depends therefore on the pose estimate) and $\boldsymbol{\beta}^{\alpha}_{k,i}$ is the bearing observation error of the detection i at timestamp k. It is supposed white, centered with a known variance.

The same process is applied to any camera that provides detections.

4.3.5 *GNSS and Dead Reckoning*

GNSS measurements are obtained at the antenna position. By combining these measurements with data from an IMU, the receiver estimates the 2D coordinates of the antenna and the vehicle's heading within a given frame G. To determine the vehicle's pose, we must apply a translation between frame B (the vehicle's main reference frame where the pose is estimated) and the antenna, known as the lever arm. This allows us to derive the following observation model:

$$ \boldsymbol{z}^{G}_{k} = \begin{bmatrix} x_{B,k} \\ y_{B,k} \\ \theta_{B,k} \end{bmatrix} + \begin{bmatrix} \cos\theta_{B,k} & -\sin\theta_{B,k} & 0 \\ \sin\theta_{B,k} & \cos\theta_{B,k} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t_x \\ t_y \\ 1 \end{bmatrix} + \boldsymbol{\beta}^{G}_{k} \tag{4.15} $$

where (t_x, t_y) is the antenna lever arm with respect to the frame B and $\boldsymbol{\beta}^{G}_{k}$ the GNSS observation noise. In next sections, depending on the dataset studied, the GNSS measurements will be provided by an SPP or a PPP-RTK solution.

As explained a gyro was available in the vehicle, providing a straightforward measurement of the yaw rate $z^{Y}_{k} = \dot{\theta}_k + \boldsymbol{\beta}^{Y}_{k}$. This gyro is a part of the system we used to estimate a reference pose during the entire drives. It is highly precise, even if some noises are observable. More details on the sensors are provided in Appendix A.

Finally, all commercial vehicles are equipped with sensors that measure the speed of each wheel, which are crucial for braking and stability assistance systems. Using these wheel speed measurements, we can derive an observation model based on the vehicle's speed and yaw rate. By applying the vehicle's geometry, as detailed by [Welte et al., 2019], we can develop distinct observation models for each wheel. The observation models are therefore as follows:

$$ \begin{cases} z^{W_{rl}}_{k} = \dfrac{2\pi}{\rho_{rl}} \left(v_k - \dfrac{\ell_r}{2}\dot{\theta}_k \right) + \boldsymbol{\beta}^{W_{rl}}_{k} \\ z^{W_{rr}}_{k} = \dfrac{2\pi}{\rho_{rr}} \left(v_k + \dfrac{\ell_r}{2}\dot{\theta}_k \right) + \boldsymbol{\beta}^{W_{rr}}_{k} \\ z^{W_{fl}}_{k} = \dfrac{2\pi}{\rho_{fl}} \sqrt{\ell^2_{rf}\dot{\theta}^2_k + \left(v_k - \dfrac{\ell_f}{2}\dot{\theta}_k \right)^2} + \boldsymbol{\beta}^{W_{fl}}_{k} \\ z^{W_{fr}}_{k} = \dfrac{2\pi}{\rho_{fr}} \sqrt{\ell^2_{rf}\dot{\theta}^2_k + \left(v_k + \dfrac{\ell_f}{2}\dot{\theta}_k \right)^2} + \boldsymbol{\beta}^{W_{fr}}_{k} \end{cases} \tag{4.16} $$

where rl, rr, fl, and fr stand for rear left, rear right, front left, and front right wheels, respectively. For each $i \in \{rl, rr, fl, fr\}$, ρ_i is the circumference of the wheel, $z_k^{W_i}$ is the wheel rotation speed, and $\beta_k^{W_i}$ is the rotation speed noise. ℓ_{rf} is the distance between the rear and front axles, ℓ_r the distance separating the two rear wheels and ℓ_f the distance separating the two front wheels.

4.4 HYBRIDIZATION OF AN SPP SOLUTION WITH CAMERA MEASUREMENTS

4.4.1 *Parameters of the filter*

To tune the filter, we used one of the available sequences. As the yaw rate measurements were provided by a fiber optic gyro, we considered them as highly accurate. The tuning was quite straightforward. Then, we focused on tuning the covariance matrices of the evolution model and of the wheel speed. For that, we used a reference pose (PPK) as GNSS source for the filter and for error computation. We did not add any camera measurement at this stage. We conducted multiple empirical tests to identify the parameters minimizing 2D errors across the entire sequence. Then, we verified our obtained parameters using the SPP solution.

Then, we tuned similarly the variance of the bearing errors and the data association parameters by incorporating the camera measurements. For simplicity, we modeled the noise of the bearings using a single variance for all measurements, assuming the same distribution. These parameters were also empirically optimized to reduce observed errors. For data association, we filtered the HD map at each timestamp to exclude all poles beyond 50 meters from the pose estimate. We defined a gating criterion, where an association is valid if $\mathcal{D}^{C}_{k,i,j} < 0.001$.

The used variances are summarized in the Table 4.2. For the GNSS fixes, we used the (variable) covariance matrices estimated by the receiver. It is important to note that this tuning approach is not optimal, and more optimal methods could have been employed. Additionally, a more complex noise model for the detections could have been defined, though it might have strayed from the assumptions of a Kalman filter. Nevertheless, we regard this tuning as acceptable since it yields interpretable results across all sequences, enabling us to assess the contributions and limitations of our detection approach. The primary focus is on evaluating the detectors' performance for localization. However, we recognize that more robust methods will be necessary for addressing real localization challenges.

4.4.2 *Tested methods*

As explained in Appendix A, we carried out a data acquisition campaigns involving multiple sequences of a similar scenario. The nominal scenario, illustrated in Figure 4.11, was followed rigorously at each acquisition, except when roadworks blocked some roads. This scenario covered all the conditions discussed in Section 1.1.

Table 4.2: *Tuned covariances used in the filter (units omitted but expressed in the international system)*

	Covariances
Evolution model	$\left[10^{-2}, 10^{-2}, 10^{-2}, 10^{2}, 10^{-1}\right].\mathbb{I}_5$
Wheel speeds	$0.273^2.\mathbb{I}_4$
Yaw rate	10^{-12}
Bearings	4.10^{-4}

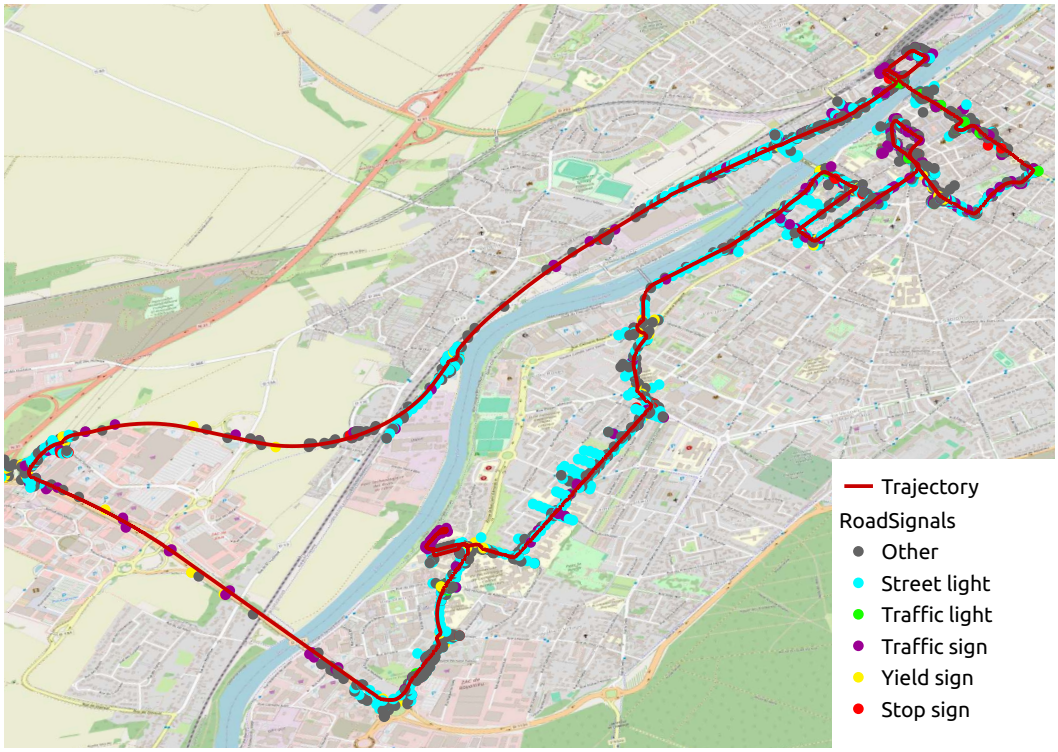


Figure 4.11: *Nominal scenario for sequences using SPP GNSS, with the trajectory shown in red. The road signs are subsampled to display only those near the trajectory.*

The GNSS computation method was SPP. To avoid significant differences in performance between sequences due to weather or traffic conditions, we chose to focus on five sequences with similar conditions, except for one with higher traffic density in some parts. The performance gap in detection methods between sequences, caused by variations in driving conditions, was presumably minimized.

To study the localization improvements, we tested the following combinations of sensors:

- **G+DR**: Uses only the INS-GNSS receiver, wheel speed sensors, and yaw rate. "G+DR" means GNSS and dead reckoning solution.
- **F**: Uses **G+DR** and the bearing measurements obtained from the front color camera.
- **All**: Uses **G+DR** and all the bearing measurements obtained from all cameras.

Additionally, as previously outlined, we had two distinct types of models with two configurations to evaluate:

- **M90**: Models trained using map annotations, with the score threshold set to achieve 90% precision for the front camera.
- **M95**: Models trained using map annotations, with the score threshold set to achieve 95% precision for the front camera.
- **MS90**: Models trained using a combination of map and segmentation annotations, incorporating uncertainty management, with the score threshold set to achieve 90% precision for the front camera.
- **MS95**: Models trained using a combination of map and segmentation annotations, incorporating uncertainty management, with the score threshold set to achieve 95% precision for the front camera.

As a reminder, the models can be ordered in ascending order of recall as follows: M95, MS95, M90, and MS90. Using the defined sensor combinations and models, we evaluated all possible pairings (corresponding to the different camera setups and perception settings), such as **F MS95** and **All M90**, which correspond to the **F** combination with the **MS95** model and the **All** combination with the **M90** model, respectively.

4.4.3 *Filter evaluation with various camera setups and perception settings*

In this section, the detected poles are matched using the vehicle's current pose estimate provided by the filter.

Figure 4.12 summarizes the 2D errors obtained for all combinations using all detection models, across all datasets, by boxplots.

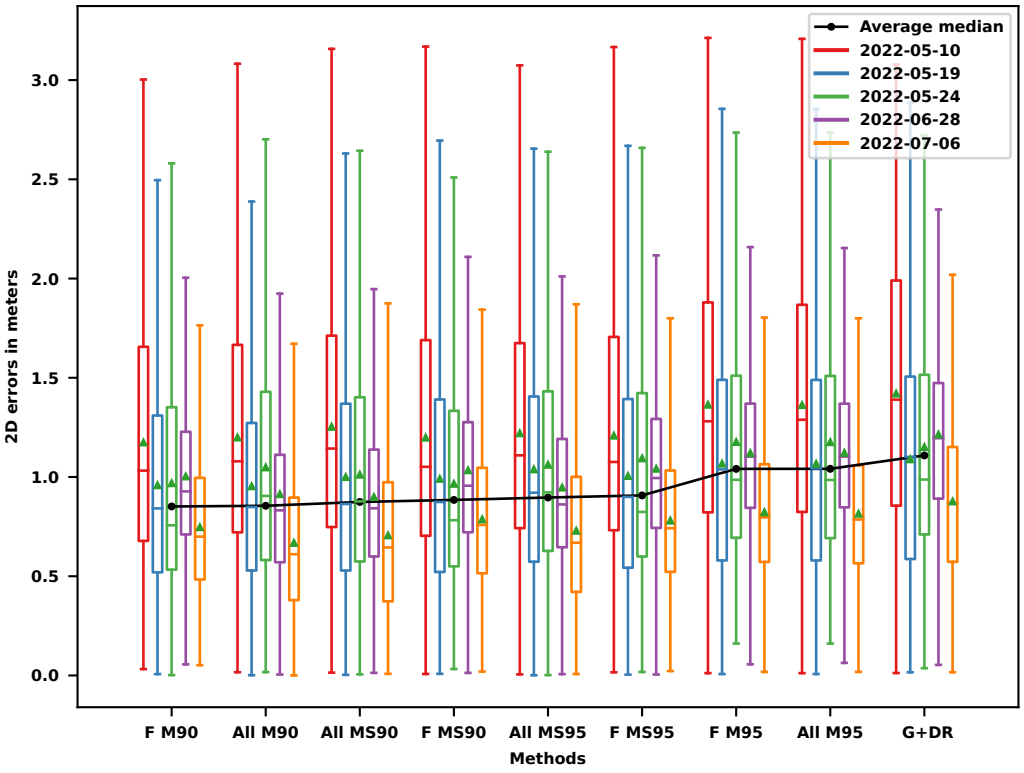


Figure 4.12: *Boxplots of 2D errors for all tested methods. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Extreme outliers have been excluded for clarity. Combinations are ordered based on the average median, emphasizing the most effective methods.*

To enhance readability, extreme errors have been excluded from the plot. Additionally, the average 2D errors for each sequence are marked with green triangles.

A first analysis of the boxplots across all sequences reveals that detecting mapped poles with cameras positively impacts accuracy. This indicates that the filter and our model perform well overall with these exteroceptive measurements, despite their challenging nature, especially for data association. It is worth noting that the intrinsic and extrinsic calibration parameters used in the filter are likely suboptimal and could be refined.

Figure 4.12 also reveals significant variability in localization performance of **G+DR** solution. For instance, the sequence 2022-05-10 shows the highest errors, whereas the sequence 2022-07-06 exhibits considerably lower errors, with both average and median errors approximately 50 cm smaller. These performance variations are also visible in the other combinations examined.

Examining all the boxplots and mean values for each pairing reveals that no single pairing stands out as superior to all the others. However, when analyzing each sequence individually, performance consistently improves with one of the pairings compared to **G+DR**, though the specific pairing that achieves the best improvement may vary and can be challenging to identify.

Typically, when considering the boxplots without the whiskers, the results are as follows: For 2022-05-10, **F M90** appears to deliver the best performance. For 2022-05-19, **All M90** seems to be the most effective. On 2022-05-24, either **F M90** or **F MS90** show the best results. For 2022-06-28, **All M90** again stands out as the best. Finally, for 2022-07-06, **All M90** continues to deliver the best results. Despite these observations, differentiating between the pairings proves difficult due to their very similar performance, particularly when focusing on median errors. When examining the mean values, they are similarly highly variable, making it challenging to visually discern the optimal pairing for each sequence based on this metric.

Furthermore, although improvements compared with **G+DR** are visible, reducing the most extreme errors proves difficult. For example, in the 2022-05-10 sequence, the whiskers remain significantly large.

Since analyzing the results from the boxplots is challenging, we propose focusing on the median errors to simplify. We calculated the average of the median errors across the five sequences for each pairing, which is illustrated by the black curve in Figure 4.12. Additionally, the methods are arranged from left to right, with the average median error increasing from the lowest to the highest.

The black curve therefore validates the accuracy improvement of all the solutions that involve vision over **G+DR**. Additionally, it highlights that **M95** provided only minimal improvement compared to the other models. This might be attributed to the notably low recall of **M95** compared with others. Finally, when focusing on this metric, the performance of other pairings appears very similar, making it challenging to rank them.

Relying exclusively on this metric can be misleading, as it may not fully capture the overall contribution of a model. For a given sequence, one pairing might

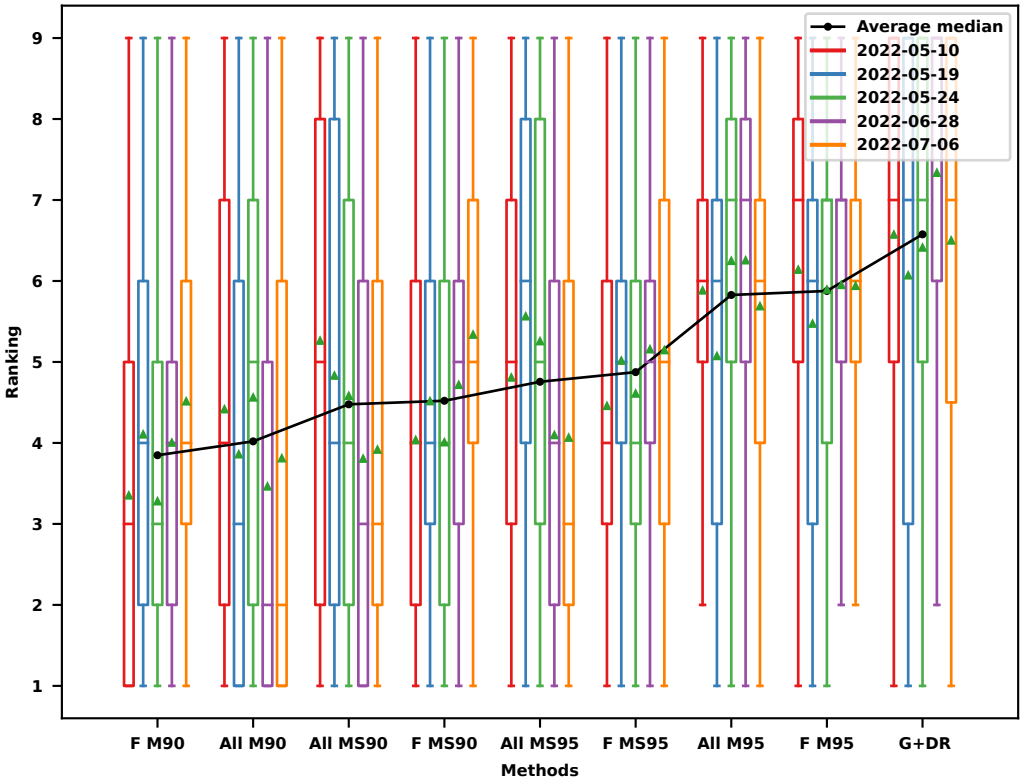


Figure 4.13: *Boxplots of rankings for all pairings. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Combinations are ordered based on the average median, emphasizing the most effective methods.*

excel in one specific area, while another one could perform better in a different segment. We would like to identify which pairing is most likely to improve performance across the entire sequence. So, we propose to use a ranking metric. For each sequence, at every timestamp, the errors produced by each method are evaluated. These errors are then ranked in ascending order, and each method is assigned the rank based on this order. If two methods have the same error, they are assigned the same rank. For instance, if we had evaluated four methods —A, B, C, and D— with errors of 0.15, 0.40, 0.10, and 0.15 respectively, the ranking would have been: 1 - C, 2 - A and D (both having the same error), and 4 - B.

Thus, for each method, we assessed the frequency of its ranks across sequences. Figure 4.13 summarizes the ranks of all combinations possible, across all datasets, using boxplots. Similar to the analysis of 2D errors, we simplified the study by calculating the average of the median ranks for each pairing as visible with the black curve.

Whether examining the boxplots or the curve, it is clear that adding cameras enhances overall performance, while the **M95** model shows more limited performance compared to the others. Regarding the other pairings, analyzing the boxplots alone makes it easier to distinguish between them compared to Figure 4.12.

Specifically, **M90** outperforms both **MS90** and **MS95** across all sequences. However, making a choice between **All** and **F**, regardless of the model, continues to be more challenging. Concerning the order of the detectors, the black curve leads to the same conclusion as before. Besides, **MS95** is also outperformed by **MS90**.

Based on the results of this case study, it does not appear crucial to achieve the highest level of detector precision. A slight reduction in precision might actually enhance overall performance. In this case, it allows detecting more mapped poles, even if there are more false positives.

However, it is surprising that **MS90** underperforms compared to **M90** for a localization task. This was unexpected, given that **MS90** provided more good detections at the same level of precision, as seen in Section 3.6. Three potential explanations are possible. First, the false positives generated by **MS90** may have been more frequently associated with map features, leading to incorrect associations and reducing localization performance. Second, the additional true positives from **MS90** might have corresponded more often to unmapped elements, resulting in poor associations. Lastly, despite **MS90**'s overall superior detection capabilities, the increased accuracy in detected pole bases positioning could have inadvertently compromised its performance during localization. Indeed, as discussed in Section 3.6, **MS** achieves superior accuracy compared to **M**, thanks to the use of annotations derived from semantic segmentation. In contrast, **M**'s automatic annotations are less accurate in the image, likely due to calibration or map errors. These calibration and map errors reappeared during the localization process when the map is projected into the C frame for data association. As a result, while correcting positioning errors during detector training may have seemed beneficial, it might have actually introduced difficulties for data association and therefore impact localization accuracy. However, further investigations are necessary to verify these hypotheses.

Finally, the addition of side cameras on the vehicle appears to have a limited positive impact with **MS95** and **MS90**. In the case of **M90**, it was actually preferable to use only the **F** rather than **All**. This seems counterintuitive, as adding more detections should generally benefit the filter. However, this could be due to a lower number of detections and an increased likelihood of incorrect associations with these additional cameras. Let us examine the 2D errors observed over time for a specific pairing and sequence, such as the 2022-05-10 with All MS95, as shown in Figure 4.14. Timestamps where camera measurements were associated with map data are indicated for all available cameras. This visualization underlines the lack of associated observations from the left and right cameras compared to the front camera. Additionally, it highlights some increases in errors at certain timestamps, likely attributable to data association issues and zones where there is no improvement of the solution. These issues also occur with other pairings and on other sequences. Finally, it underscores that, in most cases, only a few detections are associated (usually fewer than two poles per image), validating the decision to use a TC scheme for integrating perception data. Having more than two detections associated in left or right images almost never happened.

Generally, localization improvements of all pairings seemed limited due to data association. Indeed, either some correctly detected mapped elements were mistakenly associated with wrong map features or some detections corresponding to unmapped elements were incorrectly associated with some map features. Moreover, it appears more challenging to accurately detect and associate mapped objects using side cameras.

4.4.4 *Pose reference as a prior for data association*

To avoid the impact of incorrect associations in the filter, we used the reference positioning system employed for evaluation as the prior pose in the data association process. By implementing this approach, we aimed to achieve ideal associations between detections and maps, assuming there are no map errors or false detections. Even in the presence of some errors, this method should significantly reduce the impact of data association on localization, thereby allowing us to reach the optimal performance possible with our various combinations.

Figure 4.15 illustrates the 2D errors obtained over time, alongside the timestamps of camera measurements that were matched with the map, using the pose employed as a reference for localization evaluation with **All MS95** on 2022-05-10.

Even with the reference pose used for data association, some errors persist, sometimes leading to large localization errors. The errors in some areas even surpass those previously observed when the filtered pose was used as a prior for data association. For example, a notable increase of approximately 35 meters is observed in the middle of the sequence. These issues can typically arise from false detections near map features in the image. For instance, as illustrated in Figure 4.16, bollards were mistakenly classified as poles and aligned with a real mapped pole in the image, despite being distant in 3D space.

This is a major limitation of a bearing-only localization approach. Nothing in our filter can prevent these association errors. It would therefore be necessary to include a Fault Detection and Exclusion phase to prevent such errors from being injected into the filter.

Nevertheless, using the reference pose rather than the filter's estimate significantly enhances localization performance throughout the entire sequence. Figure 4.17 highlights this by summarizing the ranks of all combinations possible, across all datasets, using boxplots. As done before, the black curve represents the average of median ranks obtained on all sequences for each pairing.

Similar conclusions can be drawn for **M95**, which shows only a modest improvement in performance compared to the other models. Next, in comparison to previous observations, whether by examining the boxplots, the mean ranks associated with each boxplot, or the black curve, it is clear that **M90** outperforms **MS90**, which in turn surpasses **MS95**, whatever the combination.

Regarding the difference between **M90** and **MS90**, it seems that, with association errors minimized here, the issue is more related to the geometric accuracy differences between the two detectors. However, further investigation is required.

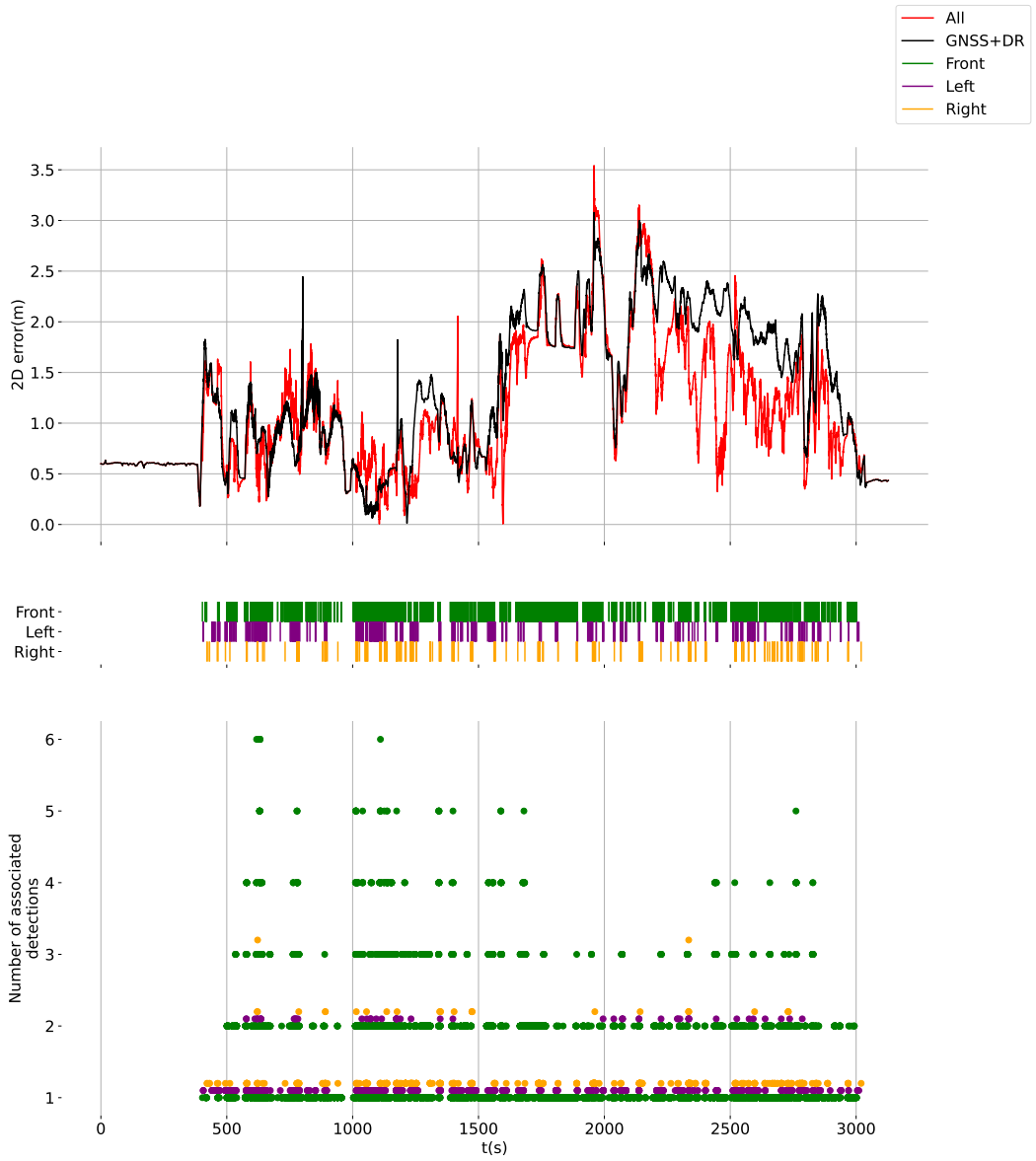


Figure 4.14: *2D errors obtained using* ***All MS95*** *on the 05-10 sequence. The observations (associated with HD map data) timestamps provided by the different cameras are summarized in the middle (front camera in green, left camera in purple and right camera in orange). At each timestamp, the numbers of detections matched with map features for each camera are visible in the bottom.*

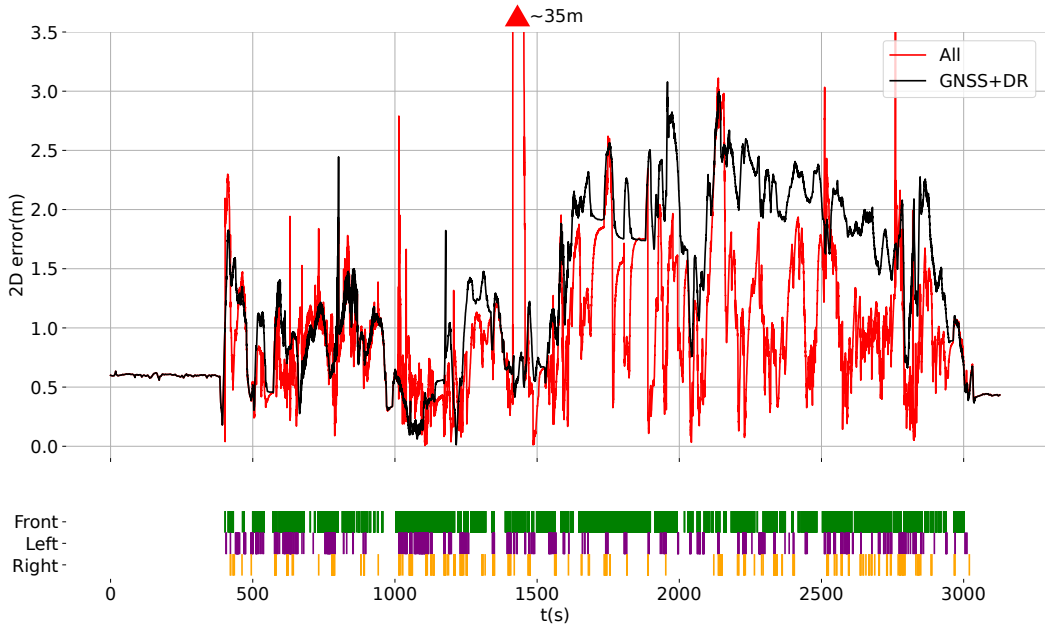


Figure 4.15: *2D errors obtained using* ***All MS95*** *on the 05-10 sequence. The observations (associated with HD map data) timestamps provided by the different cameras are summarized in the bottom (front camera in green, left camera in purple and right camera in orange). The red triangle highlights a significant increase in 2D error within the area of the graph it covers.*

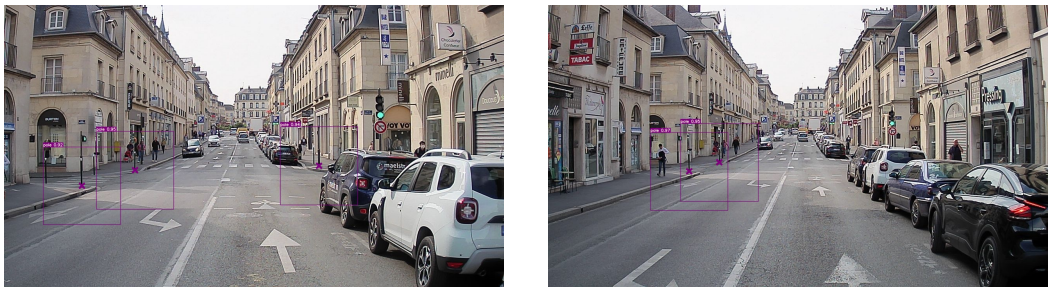


Figure 4.16: *Examples where bollards are detected as poles and perfectly aligned with mapped poles and consequently incorrectly associated with.*

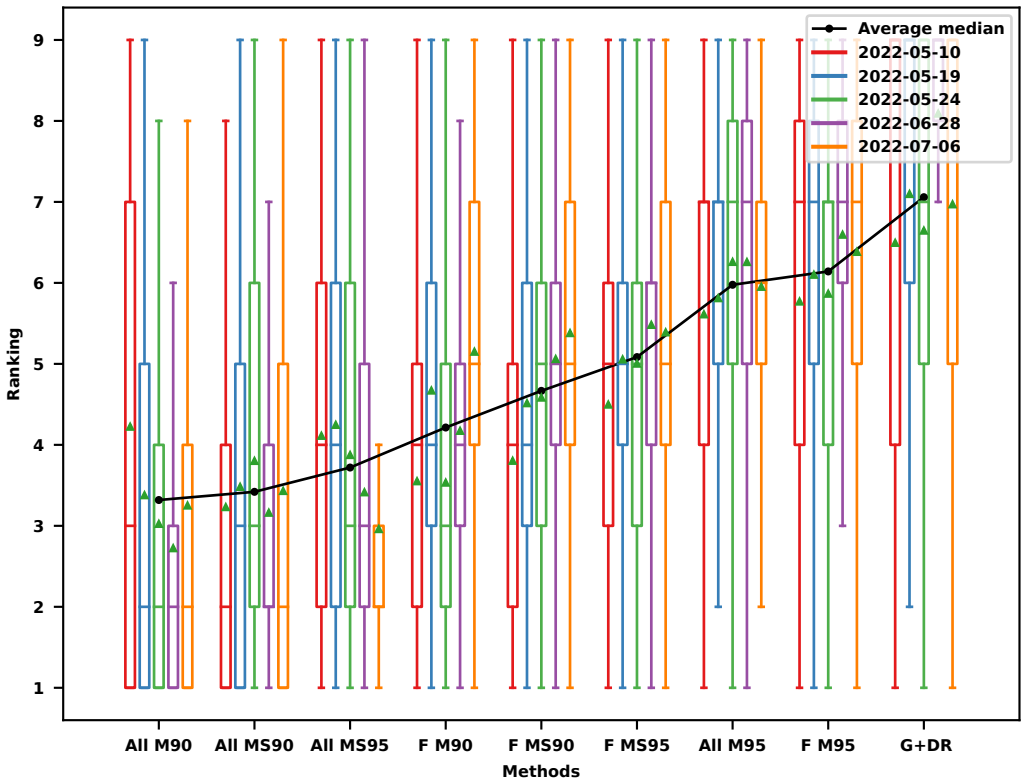


Figure 4.17: *Boxplots of rankings for all pairings. For the data association step, the reference pose used for positioning evaluation is used instead of filter estimate. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Combinations are ordered based on the average median, emphasizing the most effective methods.*

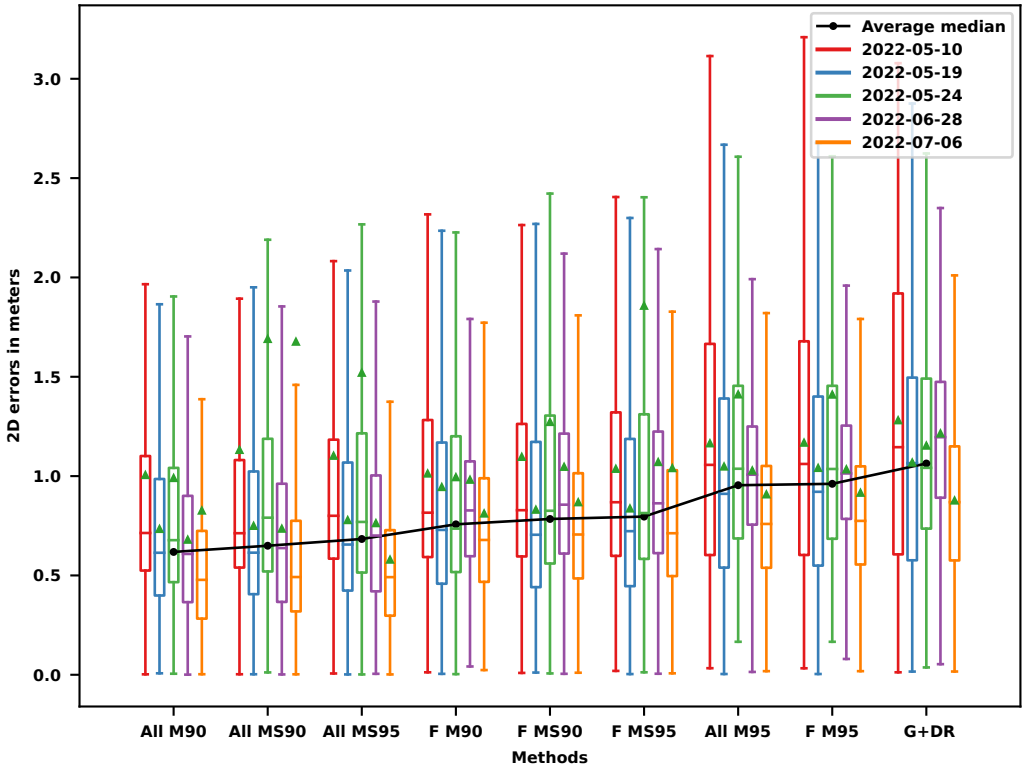


Figure 4.18: *Boxplots of 2D errors for all pairings. For the data association step, the reference pose used for positioning evaluation is used instead of filter estimate. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Extreme outliers have been excluded for clarity. Combinations are ordered based on the average median, emphasizing the most effective methods.*

Finally, unlike previous results, it is clear that adding extra cameras enhances performance for all models, as evidenced by the consistent superiority of **All** over **F** when minimizing data association issues. Here, **All M90** emerges as the optimal solution, validating the earlier assumptions about the challenges in data association. To bridge the gap between the previously observed and current performance levels, further improvements to data association and filtering methods are needed.

To conclude, the optimal 2D error results achievable with the current filter, when minimizing association errors, are summarized in Figure 4.18. As previously, boxplots display the 2D errors for all combinations across all sequences, while the black curve provides a summary by representing the average of median errors observed for each pairing. The conclusions from this figure are the same as those previously drawn from Figure 4.17. Besides, regardless of the model used, the optimal performance indicated by the average median error is remarkably better than what is shown in Figure 4.12. Typically, the filter achieves an average median error of approximately 60 cm using the optimal model, **All M90**, whereas previously the average median error was around 80 cm. Thus, although the boxplots suggest that this improvement is generally applicable across all sequences, validating this

visually remains more complex. Moreover, when looking at specific sequences, the mean 2D error indicated by the green triangle reveals that using the reference pose for data association can worsen the mean 2D error for certain sequences, as for example for 2022-07-06 with **All MS90**. This is probably due to unsolvable data association issues observed in Figure 4.15.

4.5 HYBRIDIZATION OF A PPP-RTK SOLUTION WITH CAMERA MEASUREMENTS

As detailed in Appendix 4.2, different GNSS solutions are available and can be integrated within a multi-sensor fusion framework. We examined before how camera measurements can enhance the performance of a GNSS using SPP computations. Here, we consider a much more accurate PPP-RTK solution, and study whether improvements are feasible using poles detections in our filter.

To explore this, as part of the European ERASMO project, we conducted a second data acquisition campaign, outlined in Appendix A. We focus on two specific datasets from that campaign.

First, as depicted in Figure 4.19, we did it on a shorter scenario than the previous one, to focus on peri-urban and open-sky areas within the mapped environment. This scenario significantly favors GNSS, as conditions are globally open-sky with no deep urban canyons. The color gradient highlights only a few areas where GNSS encounters difficulties, such as under bridges indicated with red rectangles, and a few areas during high acceleration and deceleration where the filter struggles to accurately estimate speed. Most of the time, the 2D errors are below 30cm which is significantly lower than the errors observed with SPP.

Figure 4.20 summarizes the 2D errors obtained for all combinations using all detection models, across all datasets, by boxplots.

As previously done, to enhance readability, extreme errors have been excluded from the plot. The data association step was done using the filter estimate. In this specific case, it is noticeable that incorporating cameras, using **MS95**, **MS90** or **M90**, results in a loss of performance. This is observed in both the boxplots and the black curve, which summarizes the means of the median 2D errors. The results suggest that the proposed filter fails to enhance performance when the initial solution is already highly accurate. Nevertheless, **M95** does not seem to have an impact on localization performance, positively or negatively. It should be remembered that these perception settings lead to cautious detection, providing few pole observations. This aligns with our earlier findings in the SPP scenario, where the enhancements provided by **M95** were minimal.

However, an analysis of the results in terms of rankings, as depicted in Figure 4.21, indicates that **M95** can enhance performance in specific sections of the scenario during both sequences compared to other models, but the improvements are relatively modest.

Enhancing the performance of an already highly accurate GNSS solution therefore appears to be challenging. This difficulty may come from the limited accuracy

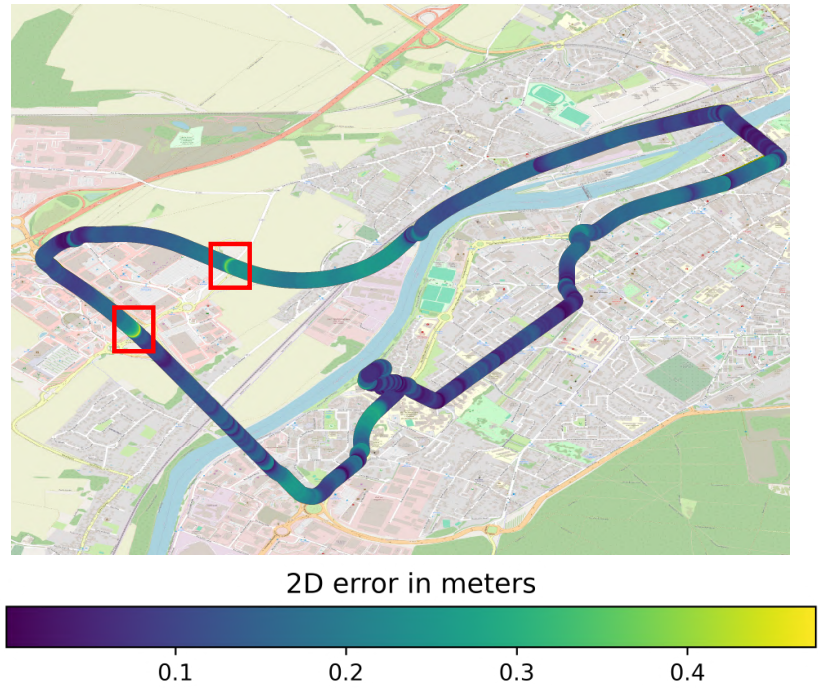


Figure 4.19: *Scenario for sequences using PPP-RTK GNSS. The trajectory shown corresponds to the path followed in the 2024-07-15 sequence. The colors represent the 2D errors of the* ***G+DR*** *solution. Red rectangles indicate situations where GNSS performed poorly because of passages under bridges.*

of the detector, mapping errors, extrinsic calibration errors or filter tuning. The combination of these issues results in a camera information that is inherently less accurate compared to the GNSS data. It is certainly possible to improve the accuracy of a camera-based system by addressing the issues mentioned above, but it seems difficult to improve them by a factor of 10. On the other hand, such an exteroceptive system can be used to supplement GNSS in places where satellite visibility is very limited or absent. It can also provide an independent localization source that can help increase the integrity of localization. In our scenario, it is preferable to either exclude cameras or choose the most effective detection model available for localization, which appears to be **M**. Using a high score threshold ensures high precision, minimizing the risk of introducing errors through data fusion.

Finally, it is important to note that these results were obtained in post-processing, not in real-time on board the vehicle. The PPP-RTK solution that was used experienced a significant latency of around 200ms, which can substantially affect localization performance. Consequently, if these latencies had been accounted for in another and dedicated filter, we might have observed different results.

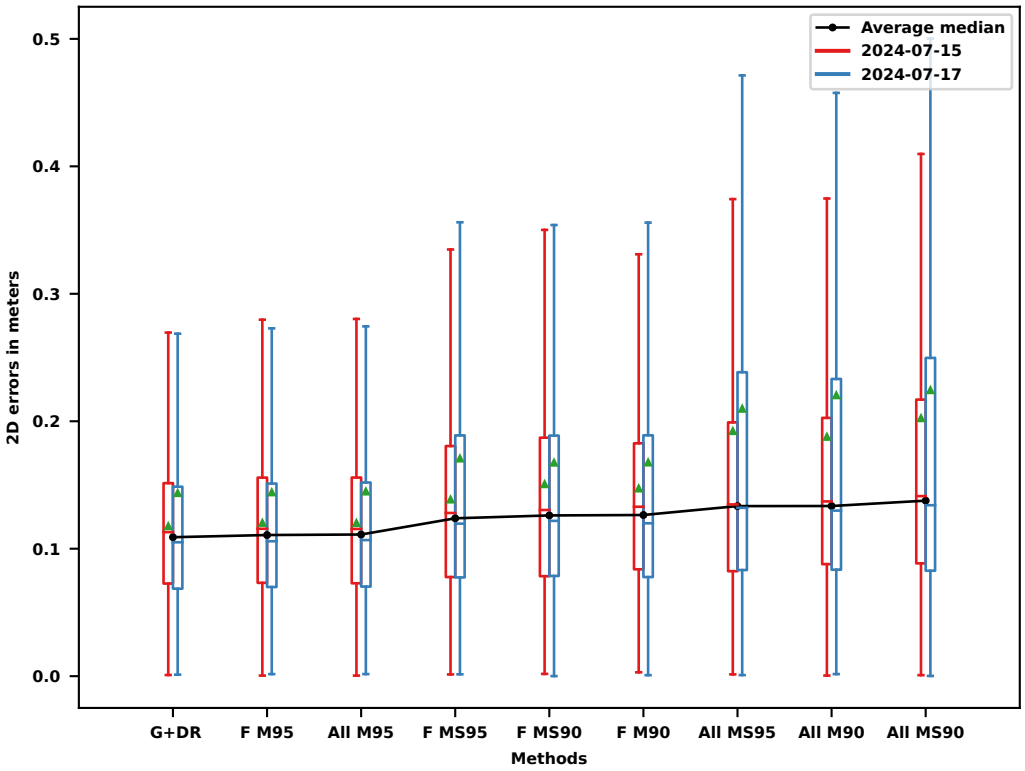


Figure 4.20: *Boxplots of 2D errors for the* ***G+DR****,* ***F****, and* ***All*** *combinations using the specified detection models. The sequences studied are those where PPP-RTK is used by the GNSS receiver. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Extreme outliers have been excluded for clarity. Combinations are ordered based on the average median, emphasizing the most effective methods.*

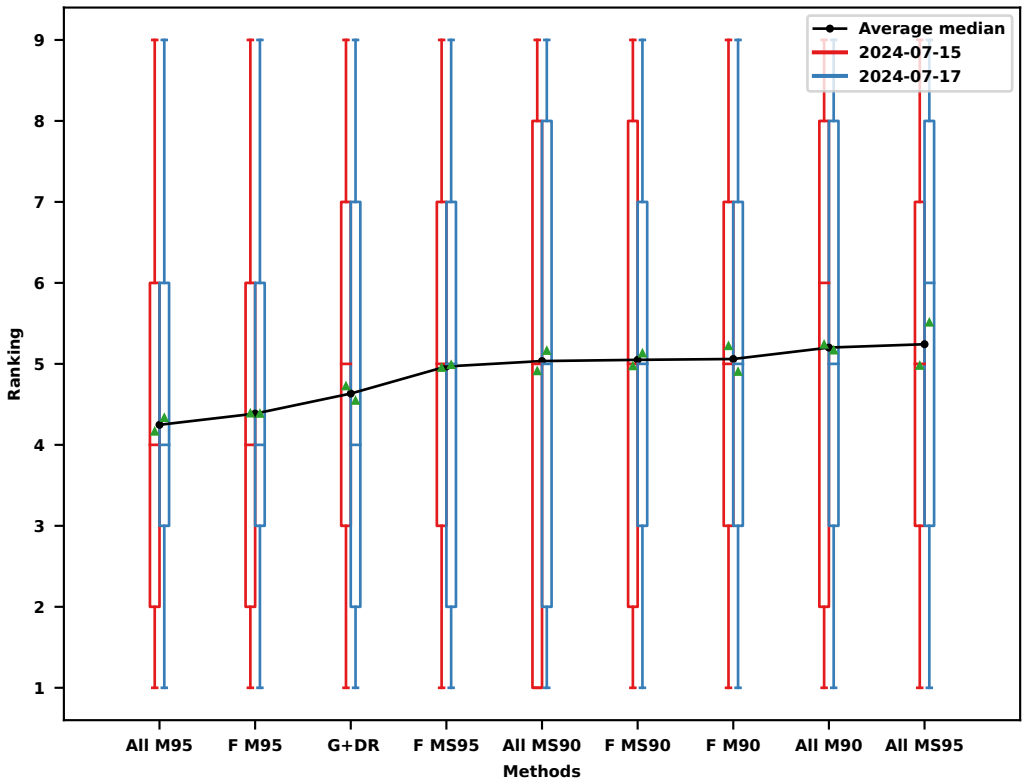


Figure 4.21: *Boxplots of rankings for the* ***G+DR****,* ***F****, and* ***All*** *combinations using the specified detection models. The sequences studied are those where PPP-RTK is used by the GNSS receiver. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Combinations are ordered based on the average median, emphasizing the most effective methods.*

4.6 CONCLUSION

In this chapter, we integrated a multi-camera system into a hybridized GNSS localization solution to enhance accuracy using the developed pole base detectors with different settings.

Our results show that integrating a map-aided multi-camera system with automatically trained detectors shows promising results for localization. Indeed, we have observed that this approach significantly enhances accuracy and improves the performance of an SPP solution over extended sequences. Additionally, our findings suggest that the score threshold parameter of machine learning-based detection algorithms has a substantial impact on localization performance. In our case study, we found that slightly lowering detection precision can be beneficial, as a robust data association algorithm, like the one employed here, is capable of managing some false positives. Nevertheless, we believe that the filter should be further robustified with mechanisms for rejecting outliers that inevitably appear. In a rather surprising way, our best detection model (trained with map-based and segmentation-based automatic annotations with black patches to manage annotation uncertainty) did not provide the best results for localization accuracy. In fact, it was surpassed by the model trained only with the map. It therefore appears that, for localization, the most effective model may not be the one with the highest detection performance. This is likely due to the fact that, for the map-based model (M), the annotations incorporate potential projection errors from extrinsic calibration, allowing the network to account for these errors in its detections which in the end help localization when using the same vehicle. Further investigation is required to fully state this.

In its current development state, our map-aided vision system struggles to enhance the filter's performance when the GNSS solution is already extremely accurate. This limitation is due to the inherent accuracy limitations of the multi-camera system and of the map. Further investigation is also needed because we have not refined our system's intrinsic and extrinsic calibrations. Moreover, additional research is needed to understand the underlying reasons for the observed performance variations.

Chapter 5

Lidar for Pole-based localization

Contents

5.1 Introduction

In the previous chapters, we examined several methods for automatically training camera detectors to identify poles, with a focus on enhancing vehicle localization. Integrating a multi-camera system has proven effective for improving localization accuracy.

However, monocular cameras present challenges, notably due to their lack of depth information. Additionally, their performance can be adversely affected by

weather conditions and lighting. Consequently, autonomous vehicles often incorporate additional sensors. Lidars, in particular, are widely used as outlined in Chapter 1. Unlike cameras, lidars can accurately measure 3D information and perform better under the aforementioned conditions, although with more limited range. Having 3D information from lidars can ease the detection of certain elements from handcrafted geometric rules. However, it can be challenging for various objects that may look similar to one another, therefore leveraging machine learning techniques to build more complex models is also interesting.

The methods presented in the previous chapters can be adapted to automatically develop a lidar pole detector. To illustrate this, we propose to train a classifier for lidar clusters that determines whether each cluster represents a pole. This approach serves as a proof-of-concept to demonstrate the feasibility of our techniques across different types of sensor data.

In this chapter, we begin with a state-of-the-art review of deep learning approaches for lidar, highlighting their strengths, limitations, and the potential advantages of automatic annotation methods. We then detail our own methods for automatic annotation of lidar clusters, leveraging map data and semantic segmentation of lidar point clouds as previously done in Chapter 2. Finally, we introduce a lidar-based pole detector and its integration into the localization filter presented in Chapter 4.

5.2 DEEP LEARNING WITH LIDAR DATA: STATE-OF-THE-ART

After having been extensively used for images, deep learning is also emerging in lidar data processing. Lidar processing is more challenging due to the characteristics of lidar point clouds data as sparse, unstructured and non-uniform. Besides, depending on the environment, the density of the point clouds varies. Similar to computer vision, researchers provided solutions for semantic segmentation of point clouds [Zhang et al., 2023] and 3D object detection [Alaba et al., 2022].

As detailed by [Zhang et al., 2023], two types of methods exist for 3D semantic segmentation: rule-based and point-based. Rule-based methods involve transforming the original point cloud into regular structures like 2D images [Milioto et al., 2019] or voxels [Maturana et al., 2015; Zhu et al., 2020], allowing conventional CNNs to process the data. However, this approach often leads to information loss and increased complexity, which can affect performance. On the other hand, point-based methods, as explored by [Ma et al., 2022; Geng et al., 2023; Zhou et al., 2023] attempt to address these challenges. Nonetheless, semantic segmentation approaches still face limitations due to computational requirements for real-time use and the limited understanding of 3D scenes, even when directly using point cloud data.

For 3D object detection, similar methods are applied as explained by [Alaba et al., 2022]: projection-based [Dong et al., 2023], voxel-based [Zhou et al., 2017] and point-based [Qi et al., 2017]. Since it consists in 3D bounding box regression

after classification, the same networks as the one used for semantic segmentation can be used as the point-based network proposed by [Qi et al., 2017].

These methods rely solely on data. Many datasets now exist with labeled lidar point clouds, such as SemanticKitti [Geiger et al., 2012] for semantic segmentation or Kitti-360 [Liao et al., 2022] and Nuscenes [Caesar et al., 2019] containing point-level annotations and 3D bounding boxes. However, suitable public datasets for specific detection tasks may not always be available.

This is particularly challenging for lidar data, given the wide variation in sensor models. Differences in sensor types and data acquisition platforms can impact the effectiveness of deep learning models trained on different datasets. For instance, dissimilarities in point cloud density, due to variations in sensor rings or angular resolution, can significantly degrade detection performance. To address these issues, domain adaptation methods, such as those proposed by [Tsai et al., 2023], are explored, where multi-modal automatic annotations techniques are employed.

Generally, to build datasets, automatic annotation methods can be used. [Dong et al., 2023] proposed an extension of their work on pole detection in range images built from point clouds where a deep neural network is trained on automatically annotated range images using their previous approach to detect poles.

Deep learning-based 3D object detection methods have notably enhanced performance. Yet, challenges remain in optimizing model speed and accuracy for real-time processing. Unlike image detection, there are no lidar models with widespread use, as each has its limitations, making the selection process more complex. Nevertheless, many users still opt for widely acknowledged and extensively used models, despite their potential inefficiencies for the specific task.

5.3 AUTOMATIC LIDAR CLUSTER ANNOTATION

In our approach, we propose to divide the detection problem into two substages by formalizing it as clustering and classification tasks. In the first stage, clusters of lidar points are grouped as individual objects using geometric proximity criteria. Then, in a second stage, they serve as object proposals to be classified as a pole or not. This framework follows a similar principle as in some image-based neural network that uses a region proposal step, e.g., R-CNN (note that it is not the case of the YOLO method).

5.3.1 *Lidar point cloud clustering*

The first step is to process the lidar point cloud in order to isolate clusters of points corresponding to potential objects of interest. We start by removing the points belonging to the ground by using the method proposed in [Jiménez et al., 2021]. It is the same method that was already used in the Chapter 2 for ground plane refinement with lidar, but instead of keeping the ground points, we now discard

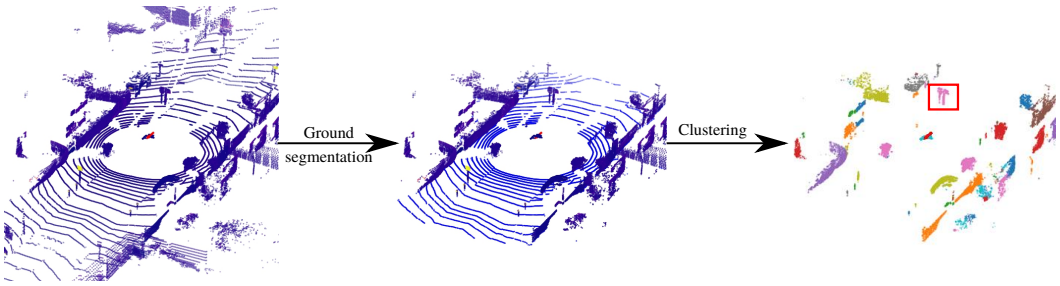


Figure 5.1: *From the point cloud provided by the sensor, ground points (in light blue in the middle plot) are removed to build isolated groups of points (clusters) corresponding to diverse objects such as poles, cars, buildings. Clusters are highlighted with various colors. During clustering, high-intensity points are removed to prevent closely spaced poles connected by a traffic sign from being merged into a single cluster. However, when traffic signs are viewed from behind, it cannot be filtered out, leading to the poles being incorrectly merged into one cluster in such case, as indicated by the red rectangle.*

them and only keep the non-ground set of points referred as ${}^{L}\overline{\mathcal{G}}$. The maximum distance D_{max} is also reduced from 50m to 20m as the point cloud becomes too sparse to compute clusters at this distance.

From the non-ground set of points, many clustering algorithms can be used to isolate individual objects. We use the method proposed in [Zermas et al., 2017]. The different stages of this process are illustrated in Figure 5.1. As highlighted, multiple poles can be merged into a same cluster because they are connected to each other through another structure (see the red square in the rightmost picture). This is typically the case for a traffic sign. To mitigate such a phenomenon, we filter out the lidar points with high intensity signal to remove reflective objects such as the traffic signs. However, this does not work when they are viewed from behind which is the case of the example highlighted in Fig. 5.1. Further processing would be necessary to decouple the two poles in this case. The output of this process is a set of clusters ${}^{L}\mathcal{C} = \{\mathcal{C}_j\}_{j=1,\dots}$, where each cluster $\mathcal{C}_j$ is a set of m_j lidar points from ${}^{L}\overline{\mathcal{G}}$.

5.3.2 *Map-based cluster annotation*

Once the clusters are computed, they need to be classified semantically as poles or non-poles. To use machine learning algorithms to classify the clusters, we need to have annotated training data with pole clusters and non-pole ones. Similar to the camera, the map data is used to generate automatic annotations.

Just like how the map poles were transformed in the camera frame in Chapter 2, they are now transferred in the lidar frame. The poles from the map are then associated with the lidar clusters. To do so, the clusters are represented as a single point by computing their average and a UNN data association strategy to match the map poles with the lidar clusters is used. We do such a matching

with a 2D Euclidean distance from a 2D bird view. All the lidar clusters that are associated to a map pole within a threshold D_{pole} are labeled as such and the rest are considered as non-poles.

5.3.3 *Lidar semantic segmentation*

The map based annotation process has the same limitation (notably missing or removed poles) as in the camera case. Therefore, we propose to compute a semantic segmentation annotation version of the lidar cluster. Just like how an image semantic segmentation neural network assigns a class to each pixel, a lidar semantic segmentation neural network assigns a class label to each individual point of the lidar point cloud. In our case, we use the Cylinder3D[Zhu et al., 2020] neural network as done in Section 2.4.2 for multi-modal image annotation. Then for each cluster $\mathcal{C}_j$, if the ratio of points within $\mathcal{C}_j$ that are classified as poles is above a certain threshold $\mathcal{P}_{pole}$, then it is considered as a pole. $\mathcal{C}_j$ is considered as a non-pole if no points are labeled as pole.

The annotation pipeline using the entire segmented point cloud is illustrated in Figure 5.2.

5.3.4 *Multi-modal annotation in a classification problem*

Contrary to the multi-modal annotation problem presented in Chapter 2 for cameras, because the two annotation methods work on the same individual cluster, the annotation fusion is from a classification perspective rather than a detection one. The combination of annotations becomes straightforward, with no need for an association phase between different sources, unlike in the case of cameras. For each cluster, we have K potential labels from K sources. We can simply decide that a cluster is a pole if there is at least k out of K methods that agree on the pole label to choose the positive examples and negative examples for the classification training. All other examples are then discarded. Since we have only two automatic annotation methods, only the union and intersection of these methods are tested.

5.3.5 *Lidar clusters annotation experimental evaluation*

To evaluate automatic lidar annotation methods, manually annotating clusters is particularly costly. This requires analyzing each cluster within its corresponding point cloud to provide context. Given the number of clusters per point cloud, which can sometimes exceed 100, this approach proves to be too expensive. Furthermore, to create annotations suitable for other detection approaches, a semantic annotation of each point cloud would be necessary.

Therefore, we propose deriving first evaluation insights from the manual annotations already conducted on the images. This approach only enables us to assess

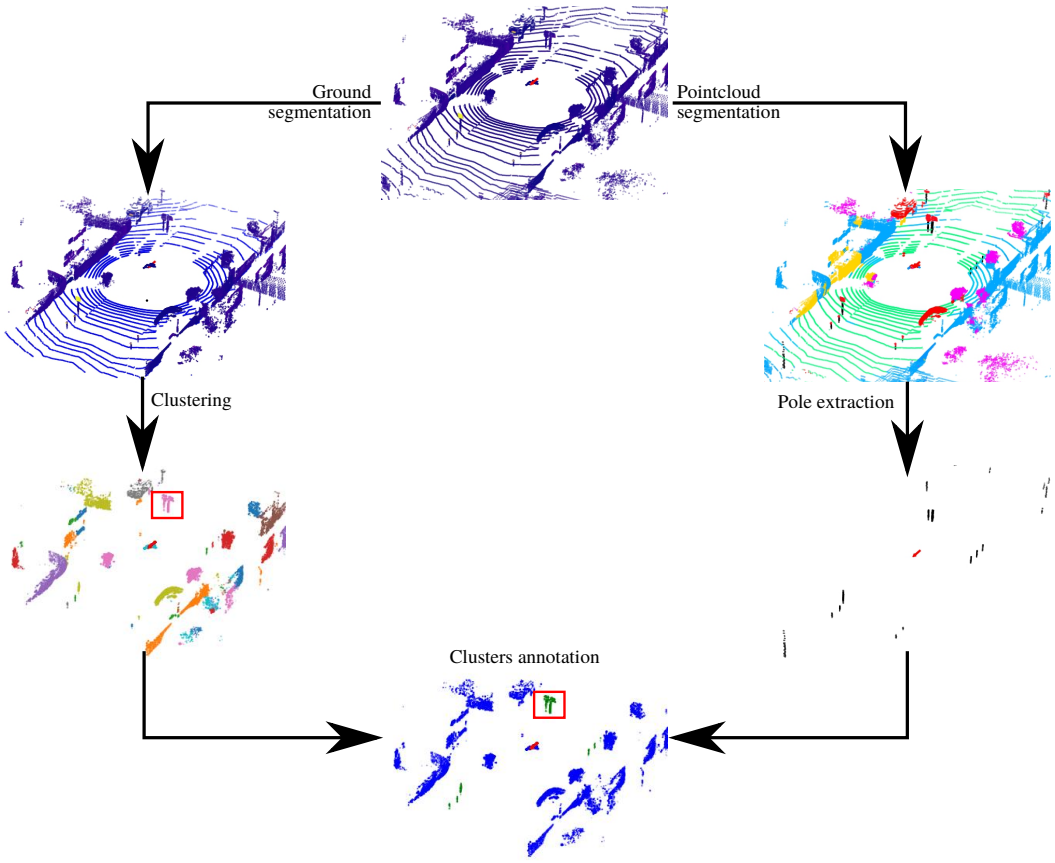


Figure 5.2: *Proposed clusters annotation using lidar data. From the original point cloud, clusters are built. In parallel, the point cloud is segmented using Cylinder3D and pole points are extracted from it as visible with black dots. Using the semantic segmentation, clusters containing a sufficient number of pole points are considered as positive examples for the upcoming classification task as visible in green. Clusters with no pole points, shown in blue, are treated as negative examples. During clustering, high-intensity points indicated in yellow are removed to prevent closely spaced poles connected by a traffic sign from being merged into a single cluster. However, when traffic signs are viewed from behind, they cannot be filtered out, leading to poles being incorrectly merged into one cluster in such case (see the red rectangle).*

Table 5.1: *Annotation evaluation of the possible strategies for lidar clusters. The clusters identified as poles are projected onto the previously used images and evaluated using the same method as image annotation. "|" and "&" indicate respectively union and intersection of annotations.*

Method	Number	FP	TP	FN	Precision (%)	Recall (%)
M	1068	503	565	8452	52.9	6.27
L	1398	898	500	8517	35.8	5.54
M & L	500	190	310	8707	**62.0**	3.44
M | L	1966	1220	746	8271	37.9	**8.17**

the annotations of clusters within the corresponding point clouds in the camera's field of view (which is limiting, as the camera's field of view is much narrower than that of the lidar, which covers 360°). To evaluate our lidar automatic annotation methods, which serve as a proof-of-concept, we project the annotated clusters onto the images and compare them with the manual image annotations, similar to the approach used to evaluate the image automatic annotation methods in chapter 2.

Following the methodology used for lidar-based annotations in images in Section 2.4.2, we project the points corresponding to the pole bases extracted from clusters identified as poles by various lidar annotation methods. As done with automatic annotations for images, to identify the corresponding manual annotation in the image for each automatic annotation, a UNN strategy with a Euclidean distance metric is applied. The results for the projected annotations obtained from the map-based method, the lidar-based method, and their combinations are summarized in Table 5.1. The dataset used is the same as those employed in Chapter 2. We automatically annotated the 2,830 lidar point clouds corresponding to the manually annotated images.

Across all methods, the recall is notably low due to the clustering's 20m distance limitation and overall clustering capabilities to identify thin clusters as poles. However, the method M achieves a precision of approximately 50% in the image projection, surpassing the lidar-based approach in precision. The lower precision for the method L is likely attributed to semantic segmentation errors and the broader definition of the pole class, similar to the segmentation-based methods for images in Section 2.4.1. L exhibits lower recall despite its potential broader definition. M seems to have a lower precision in the image projection than M method for images. This could be due to the clustering approach used.

As seen in the case of images, the union of the methods improves slightly the recall and the intersection improves the precision reaching 62%.

The total number of clusters identified as poles and the total number of clusters in the 2,830 lidar point clouds are summarized for each method in Table 5.2. The union of M and L provides a count equivalent to the maximum number of clusters identified by either method. L has fewer clusters due to discarded ambiguous

Table 5.2: *Total number of identified poles for each annotation method. "|" and "&" indicate respectively union and intersection of annotations.*

Method	Identified poles	Total of clusters
M	5080	279298
L	7182	273563
M & L	2454	267265
M | L	9808	279298

clusters with few pole points. The number of pole points in these clusters is less than $\mathcal{P}_{\text{pole}}$, but not zero, so they are not considered as either positive or negative examples. The intersection method yields the lowest number of clusters since both methods must agree.

Each method provides over 260,000 clusters, illustrating the challenge of providing manual annotations. Besides, the number of identified poles is significantly lower than the total number of clusters, resulting in fewer positive examples compared to negatives, which is expected given the observed road environments.

5.4 REAL-TIME POLE DETECTION WITH LIDAR

As explained previously, the pole detection that we propose first builds a set of clusters of potential poles, which are then classified as poles or not. Contrary to images, where hand-crafted features are no longer used, the geometric nature of lidar point clouds is well adapted to custom feature extraction. This enables classification using expert rules or machine learning. We compare both of these approaches under the hypothesis that no human-labeled lidar clusters annotation is available.

5.4.1 *Features extraction*

To this purpose, we choose to apply the Principal Component Analysis (PCA) to extract the main characteristics of each cluster $\mathcal{C}_j$. This technique allows reducing the cluster to a set of uncorrelated variables called principal components that capture the variance of the points of the cluster.

Each cluster $\mathcal{C}_j$ can be represented as a set of m_j geometric points:

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_{m_j} \\ y_1 & y_2 & \dots & y_{m_j} \\ z_1 & z_2 & \dots & z_{m_j} \end{pmatrix} \tag{5.1}$$

From this set of points, the mean point $\bar{X} = [\bar{x}, \bar{y}, \bar{z}]^\top$ is computed to deduce an unbiased estimation of the covariance matrix characterizing the point dispersion:

$$Cov = \frac{(X - \bar{X}) \cdot (X - \bar{X})^T}{m_j - 1} \tag{5.2}$$

From this matrix, we compute the eigenvalues and eigenvectors of the covariance matrix using an eigen-decomposition:

$$Cov = Q \cdot \Gamma \cdot Q^{-1} \tag{5.3}$$

$$Q = [v_1, v_2, v_3] = \begin{pmatrix} v_1^x & v_2^x & v_3^x \\ v_1^y & v_2^y & v_3^y \\ v_1^z & v_2^z & v_3^z \end{pmatrix} \tag{5.4}$$

$$\Gamma = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \tag{5.5}$$

The obtained eigenvectors v_1, v_2, v_3 indicate the direction of the components, i.e., the principal directions of the dispersion. The eigenvalues $\lambda_1, \lambda_2, \lambda_3$ indicate the amount of variance captured by each principal component, while the eigenvectors indicate the direction of these components. After that, the eigenvalues are sorted in descending order $\lambda_{(1)}, \lambda_{(2)}, \lambda_{(3)}$.

Using PCA we can describe any cluster of points using geometric properties. Firstly, we normalize the eigenvalues:

$$e_1 = \lambda_1/(\lambda_1 + \lambda_2 + \lambda_3)$$
$$e_2 = \lambda_2/(\lambda_1 + \lambda_2 + \lambda_3)$$
$$e_3 = \lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$$

Inspired by [Alexiou et al., 2022] and CGAL library, based on these normalized values, we compute geometric characteristics comprised between 0 and 1.

- Linearity: $e_1 \gg e_2$, $e_1 \gg e_3$

$$linearity = (e_1 - e_2)/e_1$$

- Planarity: $e_1 \approx e_2$, $e_1 \gg e_3$, $e_2 \gg e_3$

$$planarity = (e_2 - e_3)/e_1$$

- Sphericity: $e_1 \approx e_2 \approx e_3$

$$sphericity = e_3/e_1$$

- Omnivariance:

$$omnivariance = \sqrt[3]{e_1 e_2 e_3}$$

- Anisotropy:

$$anisotropy = \frac{e_1 - e_3}{e_1}$$

- Eigenentropy:

$$eigenentropy = \sum_{k=1}^{3} -e_k log(e_k)$$

- Surface variation:

$$variation = e_3$$

We can also define other metrics based on eigenvectors as verticality, characterizing angle difference between z-axis and the eigenvector of the highest eigenvalue:

$$verticality = \left| \frac{\pi}{2} - angle\left(v_z, v_{(1)}\right) \right|$$

Using these values, we summarize a cluster $c_{i,j}$ by using a vector of descriptors $D_{i,j}$ containing $\min x_k$, $\bar{x}$, $\max x_k$, $\min y_k$, $\bar{y}$, $\max y_k$, $\min z_k$, $\bar{z}$, $\max z_k$, $\min i_k$, $\bar{i}$, $\max i_k$, $\sum_{k=1}^{3} \lambda_k$, m_j, and the linearity, planarity, sphericity, omnivariance, anisotropy, eigenentropy, variation and verticality values.

5.4.2 *Rule-based classification*

The features described previously encode geometric information about the cluster. Because poles have a very particular 3D shape, one can try to handcraft a decision rule to distinguish pole clusters from the rest. For example, poles are typically linear and vertical structure with a significant height and limited thickness. In [Noizet et al., 2023], we proposed to consider the linearity $\mathcal{L}$, the verticality $\mathcal{V}$, the height $\mathcal{H}$ and the thickness $\mathcal{T}$ to do the classification along with the following decision rule :

$$(\mathcal{L} > \mathcal{L}_{min}) \,\&\, (\mathcal{V} < \mathcal{V}max) \,\&\, (\mathcal{H} > \mathcal{H}_{min}) \,\&\, (\mathcal{T} < \mathcal{T}_{max}) \tag{5.6}$$

where $\mathcal{L}_{min}$, $\mathcal{V}max$, $\mathcal{H}_{min}$, $\mathcal{T}_{max}$ are parameters to be set manually.

The decision rule defined previously (Eq. (5.6)) could have been learned by a decision tree from the feature vectors describing the clusters.

5.4.3 *Machine learning classification*

Instead of using a rule-based classifier, a solution is to exploit the annotations computed by the automatic annotation pipeline along with a supervised learning method. In this research, we use random forest classifiers, which add robustness compared to decision trees, but any other machine learning methods could be used for this step.

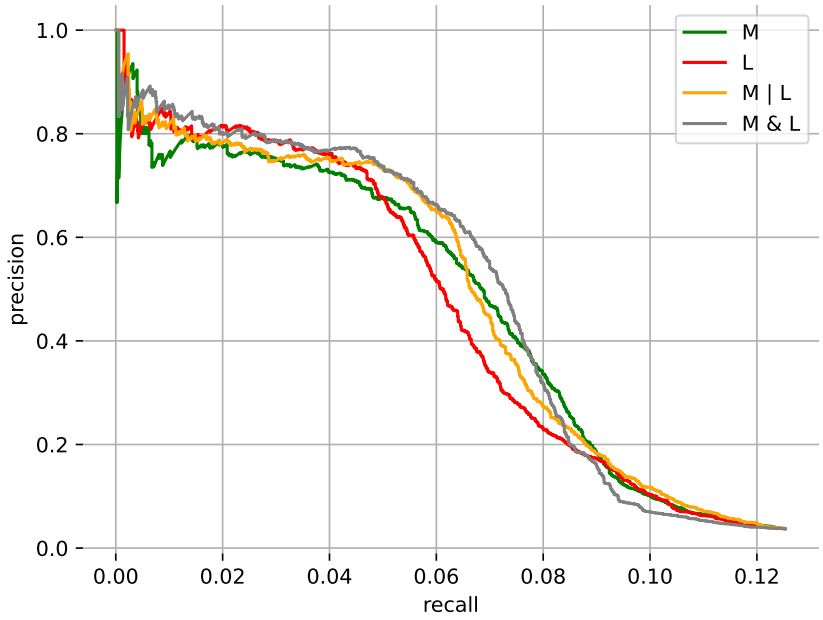


Figure 5.3: *Precision-Recall curves obtained after random forest training with different automatic annotation methods of clusters. The evaluation is realized by projecting the clusters in the image frame and using the manual annotations of the 2830 images.*

In fact, a random forest employs multiple decision trees. In a decision tree, nodes represent tests conducted on descriptors, while edges denote potential outcomes of these tests leading to subsequent nodes. Each decision tree within the Random Forest is trained on a subset of the data, and provide independent predictions. The rules applied in each tree differ, using varied tests. The final prediction is determined through a majority voting mechanism across all trees. However, a probability of being a pole or not can be returned taking into account all the votes, providing a score. This approach enhances accuracy and mitigates overfitting compared to a single decision tree.

5.4.4 *Experimental detection results*

To evaluate the presented cluster classifiers, we propose projecting the point considered as a potential pole base from each cluster onto the images and comparing these projections with the manual image annotations as described in Section 5.3.5. When classifying a newly observed cluster, the random forest estimates the probability of this cluster being a pole. Using this probability as a classification score allows us to draw PR curves similar to those generated for the camera pole base detectors.

The obtained PR curves with the different automatic annotation methods and their combinations are visible in Figure 5.3.

The intersection of M and L seems to improve the general performance except for highest recall values where the precision values obtained are the smallest.

The union is less interesting with a limited improvement of the precision for the highest recalls and a lower improvement than the intersection for recall values between 0.04 and 0.065.

As with the evaluation of the annotation, it is important to note that the performance is constrained by the clustering approach, which achieves a maximum recall of approximately 0.125 in the images. This limitation arises from the difficulty in accurately clustering poles and the inability to detect distant poles.

Note that we did not compare directly the detection perform of the rule-based classifier as it does not provide scores to be able to draw the precision-recall curve. It will be compared from a localization point of view in the following section.

5.5 POLE-BASED LOCALIZATION WITH LIDAR

For pose estimation, we use the same Kalman filter as defined in Chapter 4 with the same parameters for the evolution model. However, instead of using a multi-camera system, we integrate here a lidar processed with different poles detectors.

5.5.1 *Considered detectors*

Similarly to our work with cameras, we developed two models for detecting mapped poles with machine learning: M, trained exclusively with map data and M & L, which has shown to enhance detection performance. We compare in the following the localization performance of these two models.

Unlike with cameras, detection performance here was not thoroughly evaluated and relied solely on manual annotations in front images. This led to a limited assessment. Here, we consequently choose to avoid a similar comparison than the one proposed for cameras with different levels of precision studied since we did not evaluate it properly on entire point clouds. Instead, we fix arbitrarily the score threshold to 0.5, corresponding to a classically done majority voting in the random forest case.

It is important to note that by arbitrarily setting the threshold at 0.5 (rather than choosing it to achieve a given precision level), the precision of the two models can vary significantly, limiting the comparison. We will only be able to determine the better-performing model at this specific threshold, without fully understanding the role of the detection performance.

Finally, we also use the rule-based detector based on PCA for which we set empirically the thresholds defined in Eq. (5.6).

5.5.2 *Lidar measurements*

Given a point cloud from the lidar L, the detection output is a set of 2D measurements

$$ {}^{L}\mathbf{Y}_k^L = \left\{ {}^{L}\mathbf{y}_{k,i}^L = \left({}^{L}x_{k,i}^L, {}^{L}y_{k,i}^L \right) \middle| \, i = 1, \dots \right\}, \tag{5.7} $$

where each measurement ${}^{L}\mathbf{y}^{L}_{k,i}$ corresponds to the 2D coordinates of the centroid of the detected cluster i expressed in frame L at time k.

As done with cameras, to create an observation model that the filter can use to update its estimate, lidar detections must be associated with mapped elements. For that, the detections and the map features need to be expressed in a common frame. We can either move the map features from the O frame to the L frame or the opposite for solving the data-association. Because there are often fewer detections than map features, the latter is less computationally demanding.

To do the association of the detected poles, the detections are projected into the O frame :

$$ {}^{O}\mathbf{Y}^{L}_{k} = \left\{ {}^{O}\mathbf{y}^{L}_{k,i} = \left({}^{O}x^{L}_{k,i}, {}^{O}y^{L}_{k,i} \right) \middle| \, i = 1, \ldots \right\}, \tag{5.8}$$

To express the detections in the O frame, they are transformed in a manner similar to that described in Section 2.3, using the estimated pose (and its covariance matrix) and the extrinsic calibration parameters.

We choose to use the Mahalanobis distance between a lidar measurement ${}^{O}\mathbf{y}^{L}_{k,i}$ and a transformed map feature ${}^{O}m_{k,j}$ to solve the association problem as defined similarly in Section 4.2 in a Hungarian method.

Once the data association step is done, the observations from the lidar are injected into update stages of the localization filter. For a pair $\left({}^{L}\mathbf{y}^{L}_{k,i}, {}^{L}\mathbf{m}_{k,j} \right)$ of a detected pole and its corresponding map feature in the lidar frame, the observation model is as follows:

$$ {}^{L}\mathbf{y}^{L}_{k,i} = {}^{L}\mathbf{m}_{k,j} + \boldsymbol{\beta}^{L}_{k,i} \tag{5.9}$$

where $\boldsymbol{\beta}^{L}_{k,i}$ is the lidar observation error of the detection i at timestamp k. It is supposed white, centered with a known covariance.

5.5.3 *Tested methods*

We tested the following combinations of sensors and detectors:

- **G+DR**: Uses only the INS-GNSS receiver, wheel speed sensors, and yaw rate.
- **Expert**: Uses **G+DR** and the lidar with the rule-based detector.
- **M**: Uses **G+DR** and the lidar with the M model trained.
- **ML**: Uses **G+DR** and the lidar with the M & L model trained.

We also studied the localization performance when the lidar is merged with an SPP GNSS or a PPP-RTK GNSS (as done for cameras).

5.5.4 *Parameters of the filter*

The parameters used are the same as the ones provided in Table 4.2. The GNSS measurements covariances are again provided by the GNSS receiver. For the covariance matrix of the lidar detections, we chose to apply the same covariance to all detections. We tuned it as done in the camera case to fix it at $0.25^2 \times I_2$. We set a maximum distance of 50m to filter the map around the pose estimate for the data association step. For the data association gating, as we used the Mahalanobis distance, we fixed the gating to 6, as described in Section 4.2.

5.5.5 *Hybridization of an SPP solution with a lidar*

We study the localization accuracy obtained on the same sequences as used in Chapter 4. To evaluate the localization performance we use the 2D errors and the rankings' metrics as defined in Section 4.4.3.

Filter evaluation with several perception settings

Figure 5.4 summarizes the 2D errors obtained for all combinations of sensors and detectors, across all datasets, by boxplots. To enhance readability, extreme errors have been excluded from the plot. Additionally, the average 2D errors for each sequence are marked with green triangles.

A first analysis of the boxplots across all sequences highlights that detecting poles with a lidar improves deeply the localization accuracy, independently of the detector used. Besides, it reveals that the automatically trained detectors reach a better performance than the rule-based one, indicating that we have successfully learned a representation of a pole thanks to all the features, which is more effective than manually defined rules. While **Expert** can introduce significant errors that may degrade the **G+DR** solution, likely due to false detections impacting localization performance, **M** and **ML** consistently enhance performance across all five sequences analyzed. Additionally, this is confirmed by the black curve, which summarizes the averages of the observed medians.

As done in Section 4.4.3, we have analyzed the rankings of the different methods to enhance the analysis on the entire sequences. Figure 5.5 summarizes the ranks of all combinations possible, across all datasets, using boxplots. Similar to the analysis of 2D errors, we have calculated the average of the median ranks for each method as visible with the black curve.

Analyzing the rankings further confirms that **G+DR** is consistently improved with the use of the lidar, as this method is always ranked at least second, regardless of the sequence. By examining the boxplots, we can see that all other methods achieve the first place in at least 25% of the entire sequences. However, **ML** and **Expert** occasionally rank last, while **M** consistently ranks no worse than third across all sequences. Moreover, the analysis of the boxplots clearly shows that **M** outperforms all other methods, consistently ranking no worse than second 75%

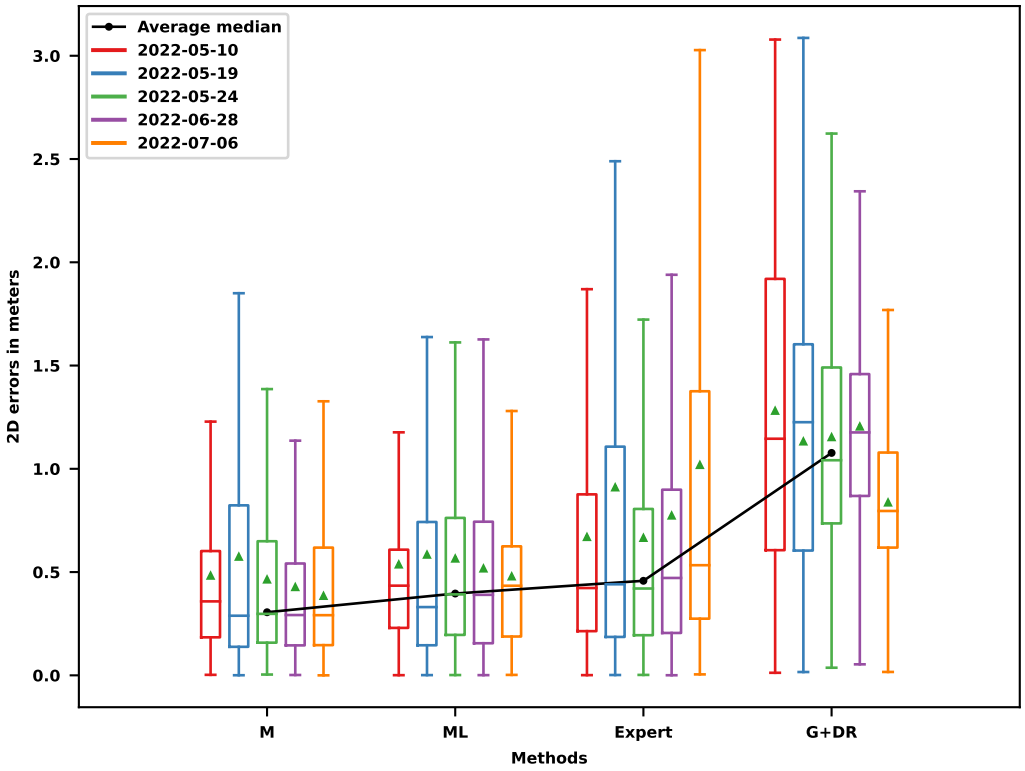


Figure 5.4: *Boxplots of 2D errors for all tested methods. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Extreme outliers have been excluded for clarity. Methods are ordered based on the average median, emphasizing the most effective methods.*

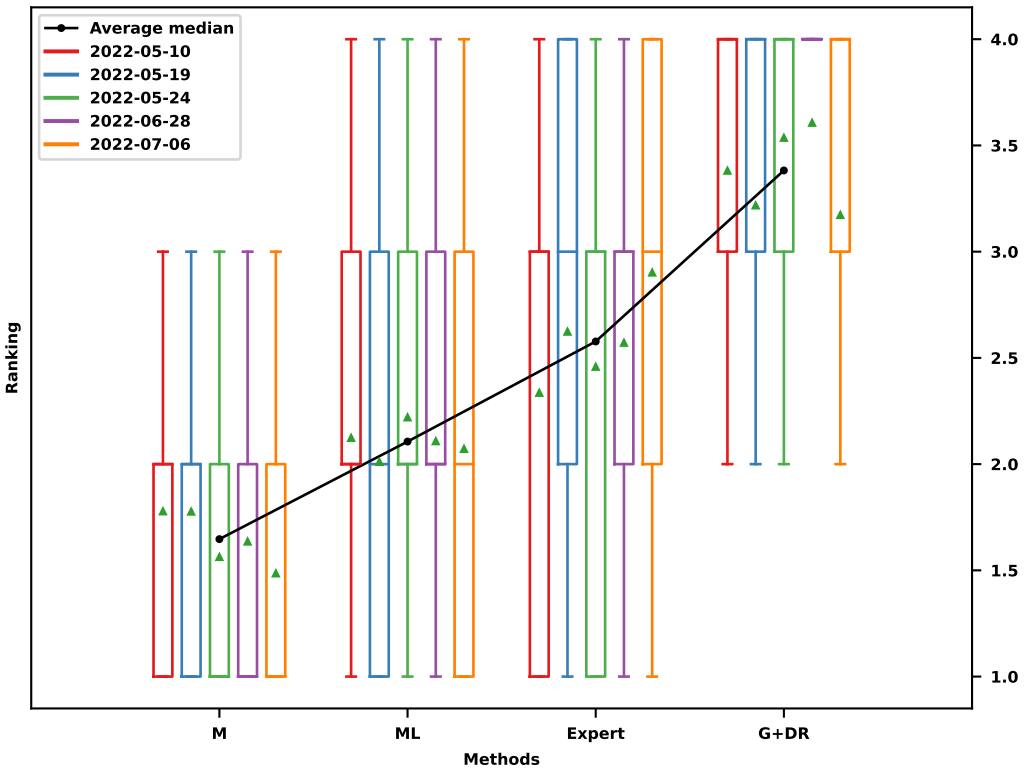


Figure 5.5: *Boxplots of rankings for all tested methods. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Methods are ordered based on the average median, emphasizing the most effective methods.*

of the time across all sequences. The average of the medians supports this conclusion and further differentiates between **ML** and **Expert**, with **ML** consistently outperforming **Expert**.

Therefore, it is clear that our machine learning approach provides substantial localization improvements. Nevertheless, **M** seems to outperform **ML**, contradicting the benefits of using a multi-modal annotation approach for training. As discussed in Section 5.5.1, comparing the two detectors is complicated because we used the same arbitrary threshold for both. With a different threshold, the results might have differed. For now, we can only speculate that in our setup, **ML** either detected fewer mapped poles, generated more false positives, or led to a higher number of incorrect associations compared to **M**.

Enhancing perception with motion compensation

One of the primary sources of association errors may come not from detection capabilities but from issues with the alignment of the lidar point clouds. Lidar motion compensation is essential for maintaining the accuracy of the captured information especially when the speed of the vehicle is high. Indeed, as lidars use a rotating platform to scan the environment, the movement of the vehicle can cause significant distortion in the acquired point cloud, especially affecting static objects. In our case, this distortion can result in incorrect positioning of the centroids of detected poles, with misalignment potentially reaching several tens of centimeters depending on vehicle speed. Such inaccuracies can lead to unassociated detections or incorrect associations with map features, ultimately impacting localization performance.

To mitigate these issues, we propose to apply motion compensation to correct the centroids positions using the mean timestamp of the points of the obtained clusters after the detection process by leveraging the kinematic data from the reference system. This alignment process ensures that the detections are now at their attended positions, thus improving data association and consequently localization. Note that in a real-world scenario, other sensors can provide the kinematic data, which might be less accurate. Thus, we are working under ideal conditions for motion compensation.

Figure 5.6 summarizes the 2D errors obtained for all combinations of sensors and detectors after applying motion compensation, across all datasets, by boxplots.

Adding motion compensation clearly improves the accuracy for all the models. After that, and as highlighted by the black curve summarizing the calculated average of 2D median errors observed, all detection models reach comparable performance in terms of localization and are hard to compare using this metric. Nonetheless, it is important to note that the performance remains impressive across all models. Specifically, the average of medians shows values consistently below approximately 20 cm for any detector, underscoring the significant advantage of incorporating a lidar pole detector for localization.

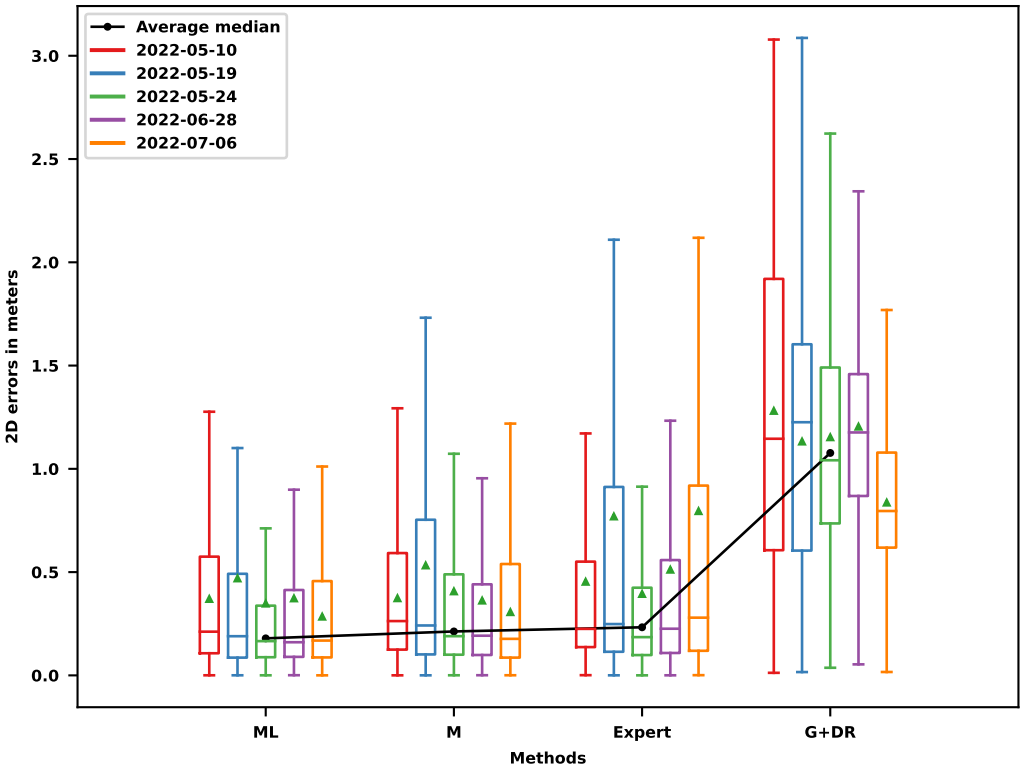


Figure 5.6: *Boxplots of 2D errors for all tested methods with motion compensation on lidar detections. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Extreme outliers have been excluded for clarity. Methods are ordered based on the average median, emphasizing the most effective methods.*

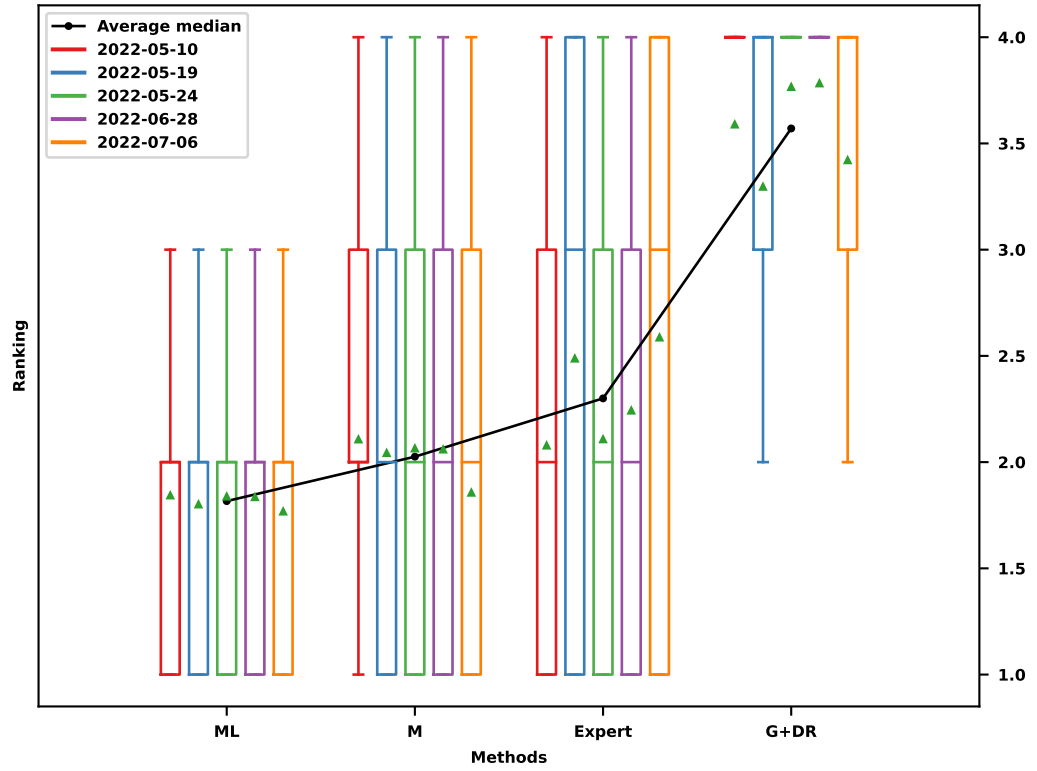


Figure 5.7: *Boxplots of rankings for all tested methods with motion compensation on lidar detections. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Methods are ordered based on the average median, emphasizing the most effective methods.*

Figure 5.7 summarizes the ranks of all combinations possible after applying motion compensation, across all datasets, using boxplots. Similar to the analysis of 2D errors, we calculate the average of the median ranks for each method as visible with the black curve.

As shown by the boxplots and the black curve, incorporating motion compensation significantly improves performance, and the baseline solution **G+DR** is clearly outperformed by the integration of a lidar sensor. Among the models, **ML** and **M** yield the best results, underscoring the value of machine learning in this context. **ML** appears to outperform **M**. However, without precise knowledge of their overall detection performance, it is challenging to determine whether **ML**'s superior results come from better training or significant differences in detection performance at the chosen score threshold. However, based on our preliminary study using manual annotations summed up by PR curves as visible in Figure 5.3, employing a multi-modal annotation method appears to enhance detection performance, which could positively influence localization results. Nonetheless, further research is required to validate these findings.

When we focus on a specific sequence, as illustrated in Figure 5.8 using **ML** on the 05-10 sequence, it becomes clear that localization performance is significantly

enhanced across the entire sequence. This improvement is largely attributed to the almost continuous detection and association of at least one pole, coupled with the frequent detection and association of three to four poles. Furthermore, depending on the pole density, it is occasionally possible to detect and associate up to 11 poles. However, this can result in a decrease in accuracy, likely due to the inclusion of elements among the 11 poles that are not actually mapped, as observed around the 2100s timestamp.

The spatial distribution of lidar detections associated with map elements during the 05-10 sequence is illustrated in Figure 5.9. Predictably, most poles are detected along the road edges, though occasional detections occur within the 20-meter radius around the vehicle, which is the maximum distance set for clustering. This underscores a key limitation in the lidar detection performance evaluation method presented previously. Since poles are detected primarily along the vehicle sides, relying solely on manually annotated images from the front-facing camera restricts the scope of the performance analysis. Note that poles on the right side are easier to detect. It is likely due to right-hand driving, making them more clearly visible.

The good associated detection distribution along each side of the vehicle, combined with the ability to regularly achieve up to three or four detections associated with the map, explains the observed increase in accuracy.

We have observed that using a reference pose obtained via PPK computation for data association (after motion compensation) does not provide any additional accuracy improvement compared to using the current filter estimate for association.

5.5.6 *Hybridization of a PPP-RTK solution with a lidar*

As done with cameras, we now study if the lidar can enhance the performance of a GNSS that uses a PPP-RTK computation. We consider the same two sequences used in Section 4.5. Since motion compensation improved a lot the performance in the SPP case, we keep it for the hybridization with the PPP-RTK solution. The data association is done using the filter estimate.

Figure 5.10 summarizes the 2D errors obtained for all methods, across all datasets, by boxplots.

No localization improvement is observed when adding a lidar using our detection models. It can even lead to a loss of performance in terms of accuracy. This is observed in both the boxplots and the black curve, which summarizes the means of the median 2D errors. An analysis in terms of rankings, as depicted in Figure 5.11, confirms the absence of positive contribution to the pose estimate. However, the trained models **M** and **ML** result in less loss of performance than the **Expert** method. With a PPP-RTK GNSS computation, the same conclusions as done with the cameras can be established.

These results underscore the significant challenge of enhancing the accuracy of a high-accuracy GNSS positioning solution, even with the use of advanced 3D

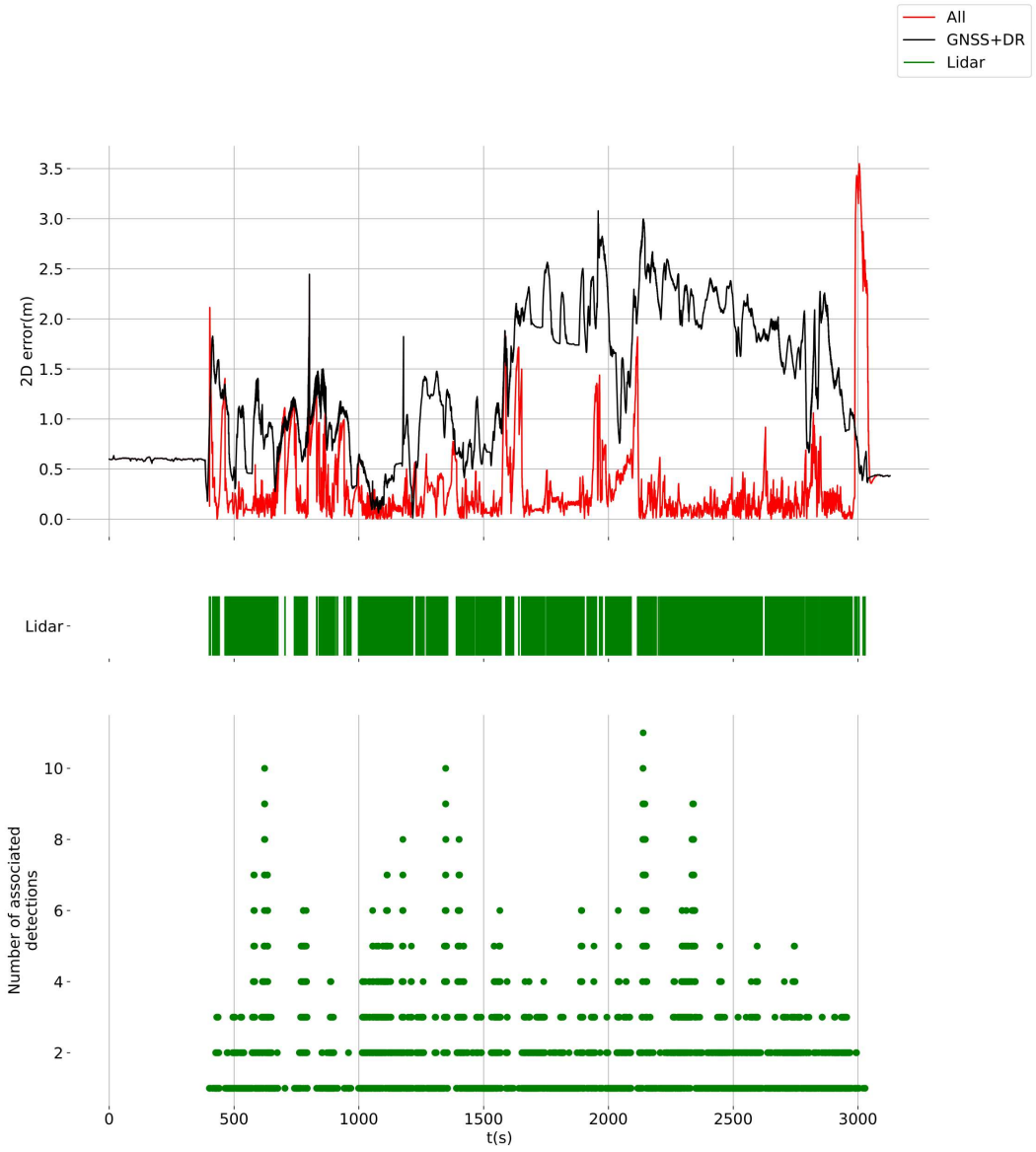


Figure 5.8: *2D errors obtained using* ***ML*** *with motion compensation on the 05-10 sequence. The observations (associated with HD map data) timestamps provided by the lidar are summarized in the middle. At each timestamp, the number of detections matched with map features is visible in the bottom.*

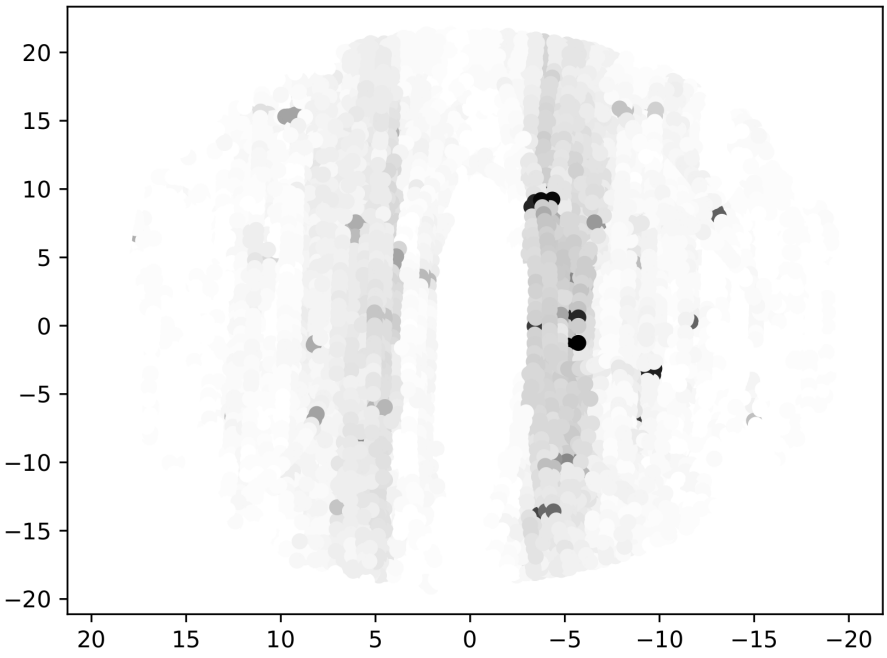


Figure 5.9: *Spatial distribution of lidar detections associated with map elements throughout the 2022-05-10 sequence. The color gradient reflects the frequency of associated detections within each area, with darker regions indicating higher occurrences of associations.*

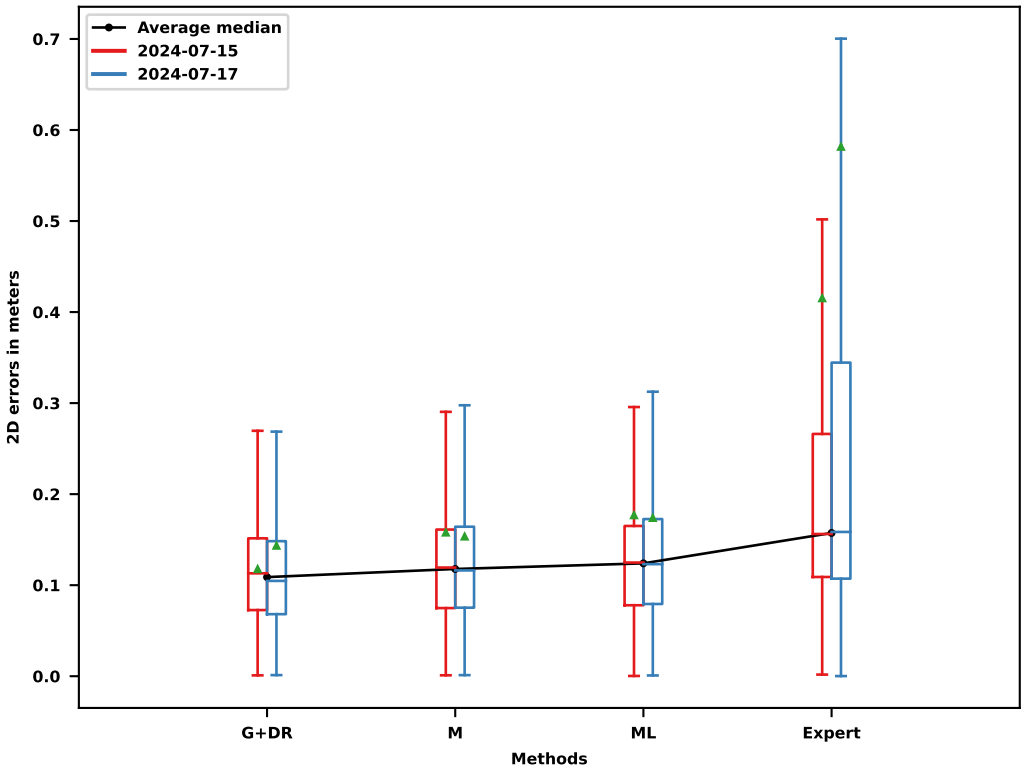


Figure 5.10: *Boxplots of 2D errors for all tested methods with motion compensation on lidar detections. The GNSS computation mode is PPP-RTK. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Extreme outliers have been excluded for clarity. Methods are ordered based on the average median, emphasizing the most effective methods.*

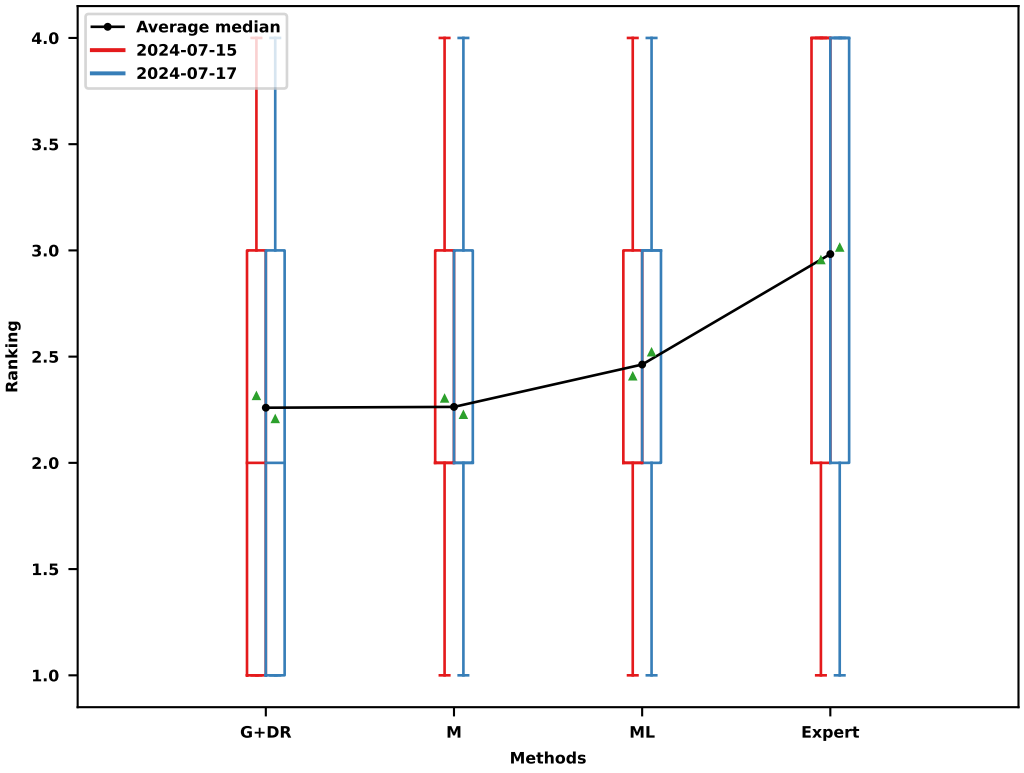


Figure 5.11: *Boxplots of rankings for all tested methods with motion compensation on lidar detections. The GNSS computation mode is PPP-RTK. For each sequence, the mean is marked by a green triangle, while the black curve represents the average of the medians across the five sequences for each combination. Methods are ordered based on the average median, emphasizing the most effective methods.*

sensors. Despite potential issues with filter parameter tuning, the difference in localization improvement between integrating with an SPP solution versus a PPP-RTK solution is substantial. However, it is important to note that these results were obtained through post-processing, without accounting for sensor latency that must be managed in real-time. Given that the PPP-RTK solution used here involves considerable latency, managing this latency in real-time could potentially lead to different conclusions.

5.6 CONCLUSION

In this chapter, we integrated a lidar detecting georeferenced poles into a hybridized GNSS localization system in order to evaluate if it can improve the accuracy.

To achieve this, we employed a methodology similar to that used for cameras. We adapted automatic map-based and lidar-based annotation techniques by introducing annotations of lidar clusters instead of directly annotating point clouds. Given the nature of the lidar data, this approach was the most straightforward way to develop a proof-of-concept. By using clusters generated through a clustering algorithm, we defined the annotations to solve a classification problem, identifying whether a cluster corresponds to a pole. This approach simplified the challenge compared to object detection in images.

The conclusion we reached when using automatic annotations and various combinations is consistent with what was observed with cameras: combining annotations increases recall, while intersecting annotations improves precision. However, our evaluation was limited by the lack of manual cluster annotations, which are very costly to obtain due to the large number of clusters across all point clouds. Instead, we relied on projecting annotated lidar clusters into manually annotated images, which proved to be limited. With its 360° field of view, lidar can detect more clusters corresponding as poles on the sides of the vehicle than directly in front of it. As a result, a better evaluation that takes into account this characteristic is needed. To address this without the high cost of annotating point clouds, we could leverage existing datasets with multiple sensors or annotate additional camera images around the vehicle to cover more of the lidar's field of view. Nonetheless, annotating a limited number of point clouds may still be considered an option.

Additionally, the capabilities of clustering lidar point clouds are inherently limited for poles detection. Clusters that are too far from the sensor cannot be obtained due to lidar range constraints. The clustering method was manually tuned, which restricts both the perception capabilities and the quality of annotations, thereby affecting the performance of the detectors.

We used these annotations with a well-known classifier (a random forest) to capture the geometric characteristics of each cluster to develop a fast detector. The learning process established geometric rules, and the resulting detectors were evaluated in a manner similar to the annotation process. The model trained using

the intersection of map-based and lidar-based annotations outperforms the one trained on map-based annotations, but further research is needed to confirm this conclusion. We compared the performance of machine learning with a rule-based detector for localization and found a clear benefit in favor of machine learning. However, our limited evaluation makes it challenging to differentiate between detectors trained exclusively on map-based annotations and those trained with both lidar and map annotations.

Incorporating lidar detections into a localization system can significantly improve the accuracy of a classical GNSS receiver. This requires effective motion compensation with a rotating lidar. Further research is needed to determine if a map-aided lidar system is beneficial when GNSS fixes are computed using PPP-RTK.

At this stage, while we have demonstrated the benefits of effective motion compensation with a rotating lidar for localization, this has not been considered in the automatic annotation methods presented in this thesis, whether for images or lidar clusters. As a result, we have had to use high threshold values to manage occlusions in the case of images or to associate map features with lidar clusters. Incorporating motion compensation would refine our approaches.

General conclusion

Contents

synthesis

HD vector maps play a crucial role in autonomous navigation by providing a priori information about the environment. These maps contain georeferenced features without requiring prior exploration by the end-user. Their lightweight design enables easy download for any new driving environment, or large areas. Additionally, the growing interest from industry suggests that these maps are likely to become widely adopted in the near future.

To use these maps for navigation, a localization system with equivalent centimeter-level accuracy is required to correctly position the vehicle on the map and make navigation decisions. If such high-accuracy localization is available, for instance through the use of RTK or PPP-RTK GNSS, it can be directly used for navigation. Otherwise, a map-aided multi-sensor localization system must be built to enhance the accuracy of the GNSS solution and even completely substitute GNSS when it is not available.

To leverage these maps for localization, it is necessary to detect the georeferenced elements. The sensor-agnostic nature of vector maps allows for flexibility in sensor choice, as long as the selected sensors can detect the mapped elements accurately. Currently, a wide range of detectors for sensors like cameras and lidars are extensively used in autonomous vehicles. However, these detectors are not always well-suited to the specific map and their implementation must meet the application's real-time constraints. When working with vector maps, high-quality detection of the map features relevant for localization is crucial. Missed detections of map features reduce the availability of exteroceptive observations, while detecting non-mapped elements complicates the process of associating detections with mapped features, potentially leading to faulty observations in the localization process.

In this thesis, we focused on detecting geo-referenced poles using cameras and lidar as a case study. It appeared that no existing detector was adequately tailored to the map we were using.

Heuristic-based detectors that rely on geometric constraints can be developed, but they are inevitably outperformed by modern machine-learning based detectors. However, these approaches demand large datasets, which are often unavailable when no existing detectors are suited for the targeted detection task. Additionally, manually annotating data is extremely costly. To address this issue, we studied how to automatically generate annotations to train machine-learning based detectors using datasets acquired in the environment covered by the vector map. Indeed, these maps offer valuable environmental priors that can be effectively used for data annotation. We developed automatic annotation methods for both lidars and cameras data using these maps and a highly accurate localization system, which is needed only for the automatic annotation phase. For the cameras, we annotated elements identified as pole bases according to their mapped positions. For lidars, we chose to annotate clusters.

This automatic annotation method has proven to be extremely promising, especially for image data, as it can accurately annotate a significant number of poles while introducing relatively few incorrect annotations. Automatic methods inherently have limitations, and errors may arise from processing issues or, more commonly, from map inaccuracies (such as poles that were removed after the map was created). It is worth mentioning that the trained detectors are adapted to the driving scenario and to the specific map used. For instance, using cameras, we adapted established object detection models to develop a highly precise pole base detector, although its recall was limited due to annotation limitation in recall. The model learned to recognize poles very effectively within the targeted mapped environment and avoid taking risk of wrong detection, resulting in a detector that is robust and well-suited for that particular scenario.

To address the recall limitations, we proposed a multi-modal automatic annotation approach that leverages both image segmentation and lidar segmentation data. These methods complement our map-based annotations effectively, particularly with the image segmentation adding multiple missed annotations, though it also introduced some false annotations due to the broad definition of pole used by the network. This approach allows us to quantify the uncertainty associated with each annotation and to create an annotation set that integrates multiple annotation methods, significantly enhancing either precision or recall depending on the combination strategy. Ultimately, taking the intersection of the acquired annotation sets results in even more precise detectors, but with more limited recall.

Nevertheless, by accounting for annotation uncertainty, as demonstrated in our camera detection work by masking image elements that may correspond to pole bases but remain uncertain, we developed detectors that significantly outperform those built solely with map-based annotations.

These detectors, which exhibit minimal errors and effectively detect a substantial number of poles, hold significant promise for localization. Our research demonstrates the considerable benefits of employing a multi-camera or lidar system for localization accuracy, even when relying on automatically learned detectors with minimal human input beyond data collection for training. Despite the

challenges associated with monocular cameras, notable accuracy improvements can be achieved. For lidar sensors, the accuracy gains are particularly striking, thanks to the rich 3D information they provide and their simpler usability for localization. Additionally, our camera-based analysis shows that a strategic trade-off between precision and recall can greatly enhance localization performance. Sacrificing little precision to achieve a significant increase in recall can be highly beneficial, underscoring the importance of defining an effective precision-recall balance. Adequate recall remains crucial to contribute to the multi-sensor system.

The experimental results from this research demonstrate that a perception system tailored to localization needs can significantly enhance the accuracy of a multi-sensor localization system. This is made possible by the use of detectors specifically designed to identify map elements. These detectors strictly adhere to the definitions used in the map's design and are developed automatically, requiring no substantial human effort, although minor detection errors may still occur.

PERSPECTIVES

The results obtained pave the way for numerous perspectives, which are presented below and organized into different sections.

AUTOMATIC ANNOTATION AND DATASETS

We have shown that leveraging automatic annotation methods to develop detectors for localization, while reducing the need for manual effort, provides robust detectors that effectively identify mapped landmarks, thereby enhancing the accuracy of the multi-sensor localization system. Nonetheless, further refinements may be necessary in the near future.

Additional annotation sources: To improve the quality of the map-based automatic annotations by identifying missing or incorrect ones, we incorporated additional sources, including lidar and image segmentation networks. Using data from different sensors allows having independent annotations. However, we can also process the same modality with different methods. For example, different segmentation networks could be used to have additional annotation sources. Pseudo-labeling could also be employed to add additional annotations after a first training. This additional information will bring more robustness to the flaws of other sources. However, a more robust combination strategy should be studied, especially if some sources could be partially correlated. Better quantifying the confidence, instead of a simple vote, should bring valuable information for the training stage.

Reliance on 3D data from a lidar: The approach we introduced in this thesis relies on the 3D information provided by a lidar to estimate the ground plane as well as estimating the occlusion of a pole base. A possibility to remove the lidar

from the required setup, is to compute 3D information from a monocular camera, it could be done with structure from motion with IMU data or with recent 3D depth estimator networks such as Depth Anything [Yang et al., 2024], or with view synthesis approaches using Nerf [Mildenhall et al., 2021] or Gaussian Splatting [Kerbl et al., 2023]. These methods are likely to be less reliable than lidar data and would require additional post-processing. Combining these 3D information with semantic information to extract the ground plane could offer an interesting alternative to lidar.

Using temporal information: In this thesis, we annotated each image or lidar scan independently of the previous or following ones. In practice, all these data are sequential by nature, and the annotation at a given time should be consistent over time. In addition, part of the data could appear ambiguous at a given time, but could be easily annotated automatically a few frames before or after thanks to the different viewpoints. Accounting for the temporal constraint in the annotation could provide valuable improvement in terms of robustness.

Datasets for detection of other landmarks or with semantics: In our case study, we demonstrated that when a detector is unavailable for landmarks stored in the map or when the map uses a different definition (e.g., bollards are not classified as poles), it is feasible to construct datasets using automatic annotations to develop detectors that align with the map's definition. This approach can also be applied to train detectors for other landmarks according to the map's specifications. For example, while our study did not include semantic information for poles, it could be extended to other features with semantic data. Many detectors for traffic signs are available today, but our map uses slightly different semantics. Developing a detector that adheres to this unique semantic definition could be useful for localization tasks. Furthermore, existing detectors may not perform well in a specific mapped context. In such cases, the map can be used to create datasets for developing detectors specifically adapted to that map and area, improving their performance by training them on highly representative contexts for the intended application.

Account for uncertainty in annotations: Each annotation method can introduce positioning errors, making it essential to quantify the positional uncertainty of each annotation. By doing so, the multimodal approach could incorporate these uncertainties, enhancing the accuracy of the overall annotations. It could be advantageous for subsequent processes, such as training models or filtering annotations with higher uncertainty before finalizing the dataset. The map-based automatic annotation method used in this research requires a system that provides an accurate localization reference. By accounting for uncertainties, it could be possible to rely on less accurate localization systems.

Evaluating the value and cost of manually annotating limited images: To evaluate our annotations and detector performance, we manually annotated 2,830 images—a time-consuming process that still left some errors. It raises the question of whether it is essential to rigorously evaluate the annotation methods themselves or if assessing the detectors' performance on limited manual annotations might suffice. For that, we must determine the optimal number of annotations needed. In our study of pole base detectors in images, we found that relying solely on automatically annotated datasets provides limited insight into the actual performance of the detectors, emphasizing the necessity of a manually annotated validation set. However, it may not be necessary to annotate as many images. A strategically selected subset could be just as effective. This consideration is particularly important for lidar detectors where minimizing manual annotation is a priority. Furthermore, a well-structured automatic approach could ease the workload on human annotators by providing pre-annotations.

POLE DETECTION WITH CAMERAS AND LIDARS

By leveraging automatic annotations, we successfully trained robust pole detectors for both cameras and lidars with no human intervention required beyond the validation phase. However, there remain opportunities for further refinement in a near future.

Enhance the study of the optimal box size for camera pole base detection: The detectors developed here are designed for object detection in images using bounding boxes. However, the performance is constrained by the size of the boxes used. In this study, we concentrated on square boxes of uniform size, regardless of their position or actual size in the image. Therefore, it would be valuable to explore the impact of varying box shapes and sizes on performance. This could involve using rectangular boxes of different dimensions based on the size of the pole in the image, and potentially incorporating the distance between the sensor and the object being automatically annotated.

Impact of black patch size used for masking ambiguous poles in images: To improve detection performance and address ambiguous pole bases in images—areas not annotated by all methods—we employed black patches. We chose an arbitrary size, assuming it would be effective in covering the pole base regardless of its position in the image. While these patches proved beneficial for learning, the optimal size remains uncertain. Moreover, depending on the pole's position in the image, varying patch sizes might be more appropriate, particularly if the pole appears very small or very large. Therefore, a more detailed study on optimizing patch sizes is necessary.

Annotation uncertainty for training: The black patch approach only quantify as binary an ambiguous from non-ambiguous example. If we refine the annotation step to better quantify the annotation uncertainty, then, these uncertainties could be taken into account at the training step. For example, with techniques as confident learning, we would manage the uncertainty of existence and even the uncertainty of positioning.

Replace IoU for evaluation in camera detectors: In object detection tasks for cameras, IoU is typically used to assess whether an object has been correctly and fully identified. Given that we are adapting these methods, IoU has been our chosen metric. However, in our context, where the pole base is not a conventional object, the center of the bounding boxes is more relevant. Therefore, for a more accurate evaluation, we could use a metric that measures the alignment between manual annotations and the centers of the detected bounding boxes. This would provide a clearer assessment of the areas of interest.

Identify the nature of false positives: In this thesis, we focused on developing robust detectors to identify specific elements as defined in the map (e.g., excluding bollards). Despite this, false positives are inevitable. It is important to determine whether these false positives are random or if they closely resemble poles in our case study. For example, do the false positives predominantly correspond to bollards? If this is the case, it may be beneficial to update the map to include them.

Exploring model alternatives for camera pole base detection: The results obtained may be constrained by the choice of model, specifically YOLOv7 in this case. Modern advancements in object detection, such as Transformer-based architectures, could provide enhanced capabilities for managing spatial relationships. Additionally, the use of bounding boxes for pole base detection has limitations, and exploring pointwise approaches, if feasible, could be beneficial. However, adopting such methods might require using less established, less documented models.

Exploring model alternatives for lidar pole detection: We focused on classifying pre-defined clusters using a manually tuned geometric method with random forest. However, it is worth evaluating whether random forest is the most effective model in our case. Alternatives, such as cost-sensitive SVMs, which address the highly imbalanced distribution of negative and positive examples, could be considered. Furthermore, given that detection is constrained by the chosen clustering approach, exploring other methods that minimize heuristic tuning could be beneficial. For example, range images might offer a promising alternative.

Detectors robustness to environmental and sensor variations: In our study, the detectors were trained on datasets with automatic annotations covering

conditions similar to those used for validation. Given this, the detectors may be overfitted to these specific conditions and could struggle with different scenarios. It is crucial to verify their robustness in new environments, such as, new cities, varying day/night cycles for cameras, or any other condition that may impact the detection, to determine if additional training is needed. In such cases, it may be beneficial to assess whether a single network trained across all conditions is sufficient or if separate networks for different conditions would be more effective. Similarly, when changing sensor models, especially for lidar, performance may degrade. Despite this, since the detectors are trained using automatic annotations, adapting to new conditions is generally feasible. Moreover, having a detector overfitted to a specific scenario is acceptable as long as the scenario is broad enough to reduce the need for managing multiple detectors.

LOCALIZATION

We have shown that automatically trained detectors can improve deeply localization accuracy. Nonetheless, further refinements may be necessary in the near future.

General improvements to the filter: At this stage, filter parameter tuning is limited and based on empirical results from a single sequence. Future improvements would involve optimizing these parameters through diverse real-world data and advanced techniques. Besides, it is important to add fault detection mechanisms to identify and mitigate error sources in particular data association errors. These orientations are necessary to strengthen the integrity of the filter outputs, enhancing trust in filter estimates.

Identify areas of interest for vision: In this study, we found that when GNSS methods are highly accurate, improving accuracy with a map-aided vision system is very challenging. This raises questions about the added value of vision in such scenarios. To address this, one could investigate how vision can enhance accuracy in more challenging environments for GNSS, such as urban canyons. Additionally, one could assess the role of vision in improving localization integrity, regardless of the GNSS solution used. Perception might be particularly valuable in specific environmental contexts, so it would be interesting to focus on identifying these areas.

Localization integrity: Ensuring localization integrity generally demands a large volume of data to minimize the risk of unmanaged positioning errors in autonomous vehicle systems. A goal is to develop techniques that allow for reliable assessments of localization integrity, despite the constraints of limited data. To address this, methods that allow for cautious or modest conclusions from limited datasets are needed. If integrity risks are very stringent, errors can be

considered as rare events. A possible approach is applying Extreme Value Theory (EVT), which helps analyze rare or extreme scenarios and assists in evaluating localization integrity under evaluable risk levels. While EVT is already used for GNSS integrity, applying it to contexts with correlated data, such as during vehicle drive, presents additional challenges.

Data association improvements: For localization with indistinguishable features, the association of detections with map features is crucial. Yet we have not fully explored the impact of different data association methods on localization performance. So far, we primarily used the hungarian method, which do not guarantee pattern matching, and can lead to some data association issues. While certain issues may be unsolvable, others could be addressed with other approaches. Analyzing the effectiveness of various data association methods within our specific context is therefore interesting to study. To enhance data association, particularly with lidar detections, we can consider aggregating detections across multiple timestamps, which can significantly improve accuracy and robustness. In contrast, camera-based systems present more challenges since detections in the camera frame are harder to propagate over time. Nonetheless, employing different data association techniques in our multi-camera system could be beneficial. By leveraging the overlapping fields of view from front and side cameras, we could define new constraints, improving data association by maintaining consistency between observations from multiple perspectives.

OTHER PERSPECTIVES

HD Map Correction/Extension: We developed robust detectors that rely heavily on HD maps to identify elements according to the map's definitions. Since these detectors detect objects that conform to the map, they can be used to extend or correct map information, or to estimate map uncertainty using highly precise RTK or PPP-RTK localization systems. By observing and verifying whether elements are present on the map, we can quantify uncertainties related to their positioning and existence. Although initial observations might be limited and insufficient to create a precise uncertainty model, methods could help approximate an uncertainty model based on the available data.

Solving calibration issues by automatic processes: An important challenge is ensuring accurate extrinsic and intrinsic calibration of all sensors. During data acquisition campaigns, sensors may shift slightly or be temporarily moved for various needs or maintenance, and some sensors might even be replaced. Tracking these changes can be highly complex, and manual recalibration is both time-consuming and prone to errors. Consequently, when acquiring data across numerous sequences, it would be interesting to implement automatic recalibration methods for the entire system.

Dataset Management: We encountered significant challenges with dataset management, including issues with sequence usability and calibration inconsistencies. Not all sequences were used due to doubts about their quality or lack of data. To improve usability, it is crucial to clean and standardize the datasets, and develop tools for easier handling. A substantial effort is needed to make the datasets more user-friendly. This includes creating tools for extracting specific scenarios, annotating useful elements, and planning for various applications.

Appendix A

Experimental setup and datasets

Contents

A.1 Experimental setup

A.1.1 *Platform*

All experiments and results presented in this thesis were obtained using experimental data collected with the Heudiasyc laboratory's vehicles, specifically designed for autonomous driving research. Throughout the thesis, multiple datasets were recorded in the city of Compiègne during three distinct acquisition campaigns, each with varying setups, particularly in terms of GNSS solutions. Some of these datasets are made available at https://datasets.hds.utc.fr/.

Experiments were conducted using two of the laboratory's vehicles: an experimental fully robotized Renault ZOE, shown in Figure A.1a, which was used for the ERASMO project integration and demonstration of fully autonomous driving

(a) *Robotized vehicle for autonomous driving tests* ***(b)*** *Equipped vehicle for data acquisition*

Figure A.1: *Vehicles available for data acquisition and experimentation*

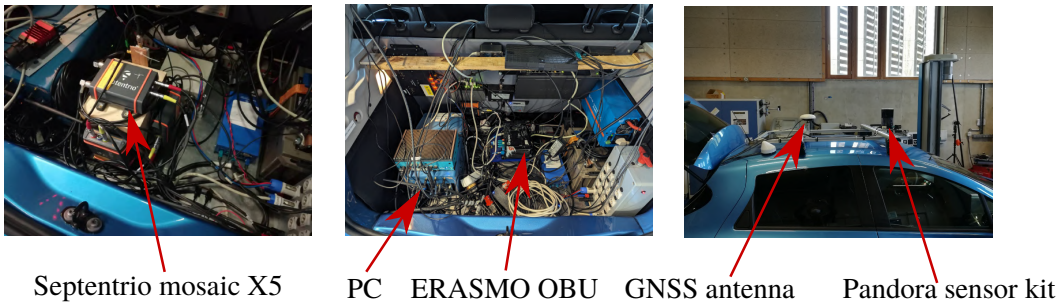


Figure A.2: *Zoom on the sensors equipped in the vehicle A.1b. The sensors used in this thesis and the PC used in the vehicle are highlighted.*

on test tracks, and another Renault ZOE, shown in Figure A.1b, which was used for data acquisition and is the vehicle associated with all results presented in this thesis.

The laboratory acquired two Renault ZOE vehicles in 2013 and 2015 through funding from the Equipex ROBOTEX (ANR-10-EQPX-44-01). These first-generation Renault ZOEs were modified to support autonomous control. Additionally, in 2019, a third-generation Renault ZOE was purchased, unmodified and solely dedicated to sensor data acquisition.

A.1.2 *Sensors*

In this section, all the sensors available in the different realized datasets are summarized, particularly those used in this thesis that are visible in the Figure A.2.

Localization reference system

All vehicles are equipped with a Novatel SPAN-CPT GNSS receiver with RTK capabilities, which is loosely coupled with a high-accuracy IMU. GNSS PPK corrections are applied using Novatel's *Inertial Explorer* software, enabling centimeter-level accuracy localization for each vehicle. This system provides the reference state used for localization evaluation. For PPK, more than ten reference stations within a 50 km radius around Compiègne are maintained by IGN, the French national

institute for geographic information. Additionally, a reference station has been deployed atop one of the university buildings, by using a Septentrio AsteRx SB PRO Connect GNSS receiver. This station can be used for PPK but is generally employed in real-time tests where high-precision localization is required by providing RTK corrections. The pose of each vehicle is provided at a frequency of 50 Hz. Additionally, all dynamic and kinematic information is fully accessible. Specifically, the IMU within this system provides the yaw rate data used throughout the results presented in Chapters 4 and 5.

CAN Bus

All vehicles provide access to their Controller Area Network (CAN) bus, allowing for the acquisition of data from internal sensors. Through this bus, gyrometer data, steering wheel angle, wheel speeds, wheel ticks, and vehicle speed can be accessed. However, there are differences between the two generations of vehicles: the steering wheel angle and wheel ticks are unavailable on the more recent model. In this thesis, since the most recent model was used, the CAN bus was primarily employed to acquire wheel speed data at 50 Hz, which was essential for the pose estimation conducted in Chapters 4 and 5.

Hesai Pandora sensor kit

This sensor is composed of a lidar delivering 3D scans of the environment using 40 vertically stacked lasers with a non-linear distribution and five monocular cameras located below the lidar. Four of these cameras are grayscale wide-angle with a horizontal field of view of 129°, providing full coverage of the vehicle's surroundings as they face the front, left, right, and rear. Consequently, both the lidar and cameras cover the same area. The fifth camera is a front-facing color camera with a horizontal FOV of 52°. The cameras are synchronized with the lidar, capturing images when the lasers align with the cameras' focal axis. Both the extrinsic and intrinsic calibrations of all sensors use factory settings. With a theoretical range of 200 meters, the system operates at a frequency of 10 Hz.

Septentrio Mosaic X5 GNSS receiver

It is a compact, low-power, triple-frequency, multi-constellation GNSS receiver, integrated in our case with an IMU to provide the vehicle's heading and improve pose estimation. Three such receivers were installed in the vehicle used in this thesis to assess the impact of IMU orientation on pose estimation performance. This contributed to the design of the ERASMO OBU, which later replaced this GNSS solution in the latest datasets. In fact, these receivers were only available in the earlier datasets from 2022, prior to the integration of the ERASMO solution. The datasets used in Chapters 4 and 5 to study the integration of our perception system with an SPP solution are part of these earlier datasets. Only one of the receivers was used in our study, operating at a frequency of 1 Hz. It was plugged to a PolaNt∗_MC Antenna.

Table A.1: *Frequencies of the different sensors used in this thesis*

	Frequencies
Septentrio Mosaic X5	1 Hz
ERASMO OBU	10 Hz
Wheel speeds	50 Hz
Reference system	50 Hz
Pandora sensor kit (cameras + lidar)	10Hz

ERASMO On-Board Unit

During the European ERASMO project, an On-Board Unit (OBU) was developed. Its purpose is to fuse data from various exteroceptive and proprioceptive sensors with an initial solution provided by a Septentrio Mosaic X5 GNSS receiver paired with an IMU. Unlike the previously introduced receivers, this one is capable of delivering a GNSS PPP-RTK solution coupled with the IMU. The OBU provides a pose estimate using the PPP-RTK solution at a frequency of 10 Hz, along with a sensor-fusion-based pose estimate at a frequency of 20 Hz. Both estimates are accompanied by integrity metrics at different risk levels, allowing for a more accurate assessment of the reliability of the localization information.

Additional sensors

Depending on the dataset analyzed, additional sensors that were not utilized in this thesis may be available. These include, but are not limited to:

Velodyne VLP-32C: This lidar sensor delivers 3D scans of the environment using 32 vertically stacked lasers with a non-linear distribution. It offers a 360° horizontal field of view and a 40° vertical field of view (from $-25°$ to $+15°$). With a theoretical range of 200 meters, it operates at a frequency of 10 Hz.

Mobileye camera: This camera provides lane marking measurements at a frequency of 37 Hz, with the capability to detect either two or four lane markings depending on the model. The vehicle (the most recent model) used in this thesis, as well as in all acquired datasets, is equipped with the four-lane model. Within the ERASMO project, this camera was used to enhance localization accuracy by associating detected lane markings with those recorded on the map.

Septentrio AsteRx SB PRO Connect GNSS receiver: This GNSS receiver was used at the laboratory before the beginning of the ERASMO project. It is a triple-frequency, multi-constellation receives capable of processing signals from constellations such as Galileo and GPS. It was configured to deliver only positioning information at 1Hz, without providing any angular data. It was plugged to a PolaNt$*$_MC Antenna.

The frequencies of all sensors data used in this thesis are summarized in Table A.1.

A.2 SENSOR CALIBRATION

As explained in the different chapters, an accurate sensor calibration is crucial for intelligent vehicles, especially when integrating data from multiple sensors to accurately model the relationship between sensor observations and the vehicle's state. This is particularly important for achieving reliable pose estimation, as demonstrated in Chapters 4 and 5. For the intrinsic calibration of all sensors, the factory settings were used, except for the Mobileye camera, not used in this thesis, calibrated a few years ago by a Mobileye engineer. For extrinsic calibration, a FARO Vantage laser tracker was employed to accurately determine the positions of all sensors relative to the body frame, centered at the rear axle. This system provides highly accurate positioning measurements, with accuracy within millimeters. The center of the rear axle was determined by measuring specific points on the rear wheels, and the position of each sensor's reference point was measured. The transformations between the vehicle's base frame and the sensor's frames were deduced. For using CAN bus data, particularly the wheel speeds used in this thesis, only distances (such as the distance between the front and rear axles and their respective lengths) and other parameters, like wheel circumference, are required, which can be obtained directly from the vehicle's technical specifications.

A.3 DATASETS

Firstly, I would like to thank Antoine Lima, Stéphane Bonnet, Thierry Monglon, Rémy Huet, Corentin Sanchez, Joëlle Al Hage and Maxime Escourrou, for their contributions to this part.

A.3.1 *Initial recordings*

Before integrating the ERASMO solution, we conducted an initial data acquisition campaign in 2022 to validate performance, particularly regarding the receiver, and to obtain usable datasets for this thesis. From the main sensors previously introduced, only the ERASMO OBU was not available. During this campaign, we performed 10 different driving sessions following the scenario described in Figure 1.1 as closely as possible. This scenario covered a range of environments, from open-sky to urban areas in a small French city, covering approximately 13 kilometers. Throughout these 10 sessions, we observed various traffic conditions and weather conditions (though limited since most of the drives were under similar conditions), and driving speeds. Additionally, we encountered roadworks during the drives.

The Table A.2 summarizes the approximate time and condition of each drive.

Examples of images during various driving sessions, illustrating some of the encountered conditions, can be seen in Figure A.4. An example of a speed profile during 2022-05-10 sequence is visible in Figure A.3 highlighting high speeds

Table A.2: *Approximate time and conditions of each drive realized during the first acquisition campaign without the ERASMO OBU*

Dataset	Time	Weather	Traffic
2022-04-06	17:00-19:00	Sunny/Rainy	Dense
2022-05-10	10:00-11:30	Sunny	Normal
2022-05-19	17:00-18:30	Sunny	Dense
2022-05-20	14:30-15:30	Sunny/Rainy	Normal
2022-05-24	09:00-10:30	Sunny	Normal
2022-06-28	10:00-11:00	Sunny	Normal
2022-07-06	10:00-11:00	Sunny	Normal
2022-07-15	10:00-11:00	Sunny	Normal
2022-09-20	09:15-11:00	Sunny	Normal
2022-09-28	14:50-15:40	Sunny	Normal

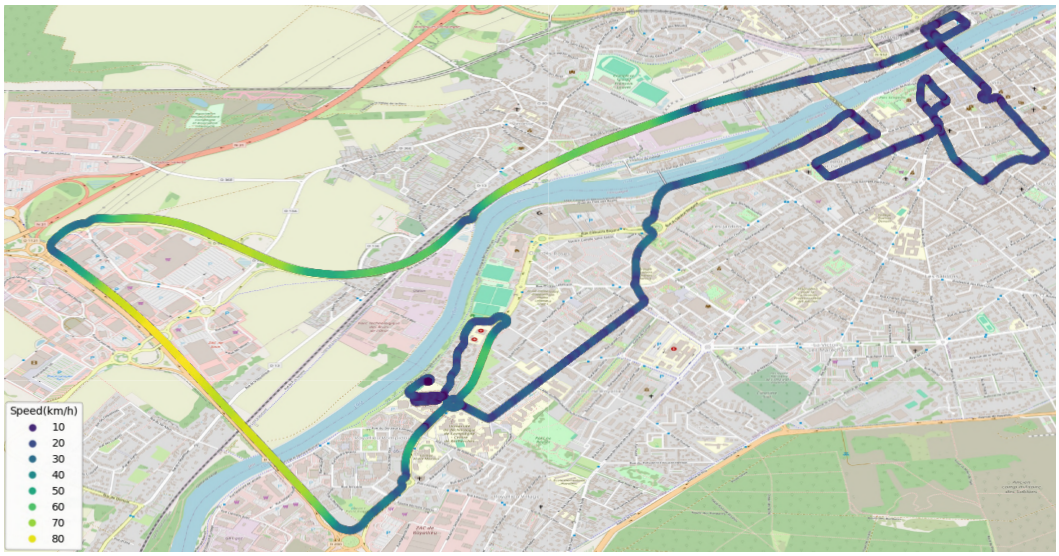


Figure A.3: *Speed profile during 2022-05-10 sequence.*

throughout a significant portion of the scenario corresponding to the open-sky zone.

Among these 10 sequences, we used sequences 05-10, 05-19, 05-24, 06-28, 07-06, and 07-15 for this thesis. Sequences 05-10 and 05-19 were used for the manually annotated image dataset, which is used in Chapters 2, 3, and 5 for evaluating the perception part. Sequences 06-28, 07-06, and 07-15 were used for automatic annotation to build our training datasets for Chapters 2 and 5. Sequences 05-10, 05-19, 05-24, 06-28, and 07-06 were used to assess localization accuracy in Chapters 4 and 5.

Other sequences were not used because they still need to be properly organized, and we need to verify the presence of all necessary data and the validity of the calibration files. This is due to several changes made to the vehicle during the data acquisition campaign.

Figure A.4: *Examples of images extracted from several datasets during the initial data acquisition campaign in 2022. These images illustrate various scenarios and situations encountered, such as heavy traffic, roadworks, stopped vehicles, urban canyons, and bridges.*

Table A.3: *Approximate time and conditions of each drive realized during the first acquisition campaign without the ERASMO OBU providing PPP-RTK computation*

Dataset	Time	Weather	Traffic
2024-07-08	17:25-18:00	Cloudy	Dense
2024-07-15	11:00-11:30	Cloudy	Dense/Normal
2024-07-17	10:45-11:15	Sunny	Normal

A.3.2 *ERASMO integration recordings*

After completing the initial datasets and validating the first GNSS-related requirements for the ERASMO project, the integration of the final ERASMO solution (including the OBU) began in August 2023. From that point until June 2024, numerous datasets (+30) were collected to assess the localization performance achieved by the OBU and to ensure the proper integration of the system. During this period, sensor positions and other major adjustments were made, which unfortunately rendered most of the datasets unusable. Not all sensors were included consistently, some datasets were very short, calibration issues occasionally arose, different vehicles were used, and some data could be missing. Despite the time invested, these datasets could not be used in this thesis or to validate the ERASMO solution's performance properly. However, they were essential to successfully integrate the final solution and validating the OBU in our vehicles.

A.3.3 *ERASMO final solution recordings*

After integrating the ERASMO solution into our vehicles, the OBU providing GNSS positioning using PPP-RTK computations became available. To study it in detail and to use an accurate initial solution for the analyses conducted in Chapters 4 and 5 of this thesis, we realized a new data acquisition campaign in July 2024, collecting three datasets. These were obtained following the scenario depicted in Figure 4.19, which is shorter than the previous one, covering approximately 10 kilometers. This scenario was selected to avoid the most challenging areas for PPP-RTK, allowing us to better highlight its performance. The observed conditions were relatively similar to those from the earlier campaign, as this scenario largely overlaps with the previous one. The approximate times and conditions for each drive are summarized in Table A.3.

However, traffic was slowed in a section of the high-speed road due to roadwork. Besides additional roadworks occurred in certain areas. While some of these works led to significant changes to certain roads, they did not affect the key aspects of our study: pole base detection and localization.

Of these three datasets, only the 2024-07-15 and 2024-07-17 sequences were used in Chapters 4 and 5 to assess whether it is possible to further improve the

Figure A.5: *Examples of images with new roadworks during the last data acquisition campaign with the ERASMO OBU.*

accuracy of a solution using GNSS PPP-RTK as the initial source in our case. The third one was not used due to the lack of wheel speeds recordings in the dataset.

A.4 SOFTWARE AND TOOLS

All developments for real-time usage carried out during this thesis used the Robot Operating System (ROS) middleware. The ROS Noetic version was used throughout the thesis, while some of the developments for the ERASMO project were implemented using ROS2 Humble. For post-processing applications or those based on use of previously acquired data, the use of ROS was avoided as much as possible to prevent dependence on the ROS data format. However, there is still room for improvement.

A.4.1 *Datasets*

Along with the previously mentioned team members, we not only recorded all the datasets presented here but also developed several tools. To facilitate dataset recording, a script was created to properly select the sensors we want to record and define a set of metadata describing the dataset. Additionally, a suite of tools was implemented to monitor the recording. To ensure the smooth execution of the data collection process, we monitor all the sensors and the PC in real-time. Furthermore, various interfaces allow real-time annotation of rare events and note-taking during the drive. Another interface helps to strictly follow a pre-defined driving scenario, functioning similarly to a standard GPS application. An example of one of the available windows during recording can be seen in Figure A.6.

Once the dataset is recorded, a set of tools allows extraction of the data in various formats, ranging from CSV for position or vehicle dynamics information, to PCD for point clouds, and MP4 or JPG for images. Other types of data can also be included. This script also reorganizes the dataset appropriately, as only a single file is generated during the drive. Finally, another set of scripts automates the

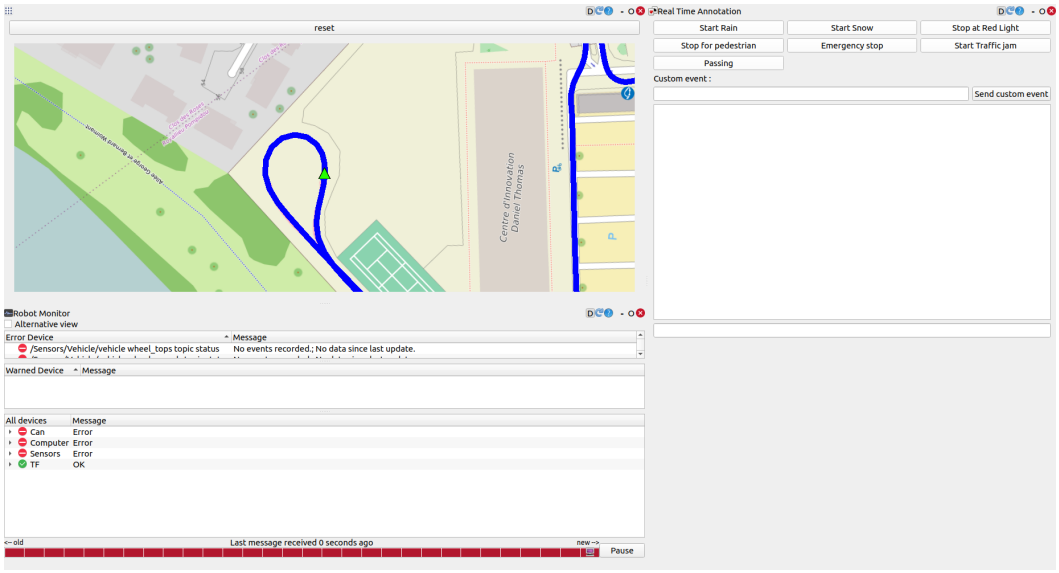


Figure A.6: *Example of an interface used during dataset recording*

extraction of key information relevant to this thesis, such as detections, without the need to work with the entire dataset.

A.4.2 *Thesis*

For this thesis, we developed a complete pipeline for automatic pole base detection training with minimal human intervention, using multi-modal automatic annotation. A similar pipeline was also created for the lidar case. Additionally, we developed a flexible library for data association and pose estimation that can handle any type of data and data association strategy. To address latency issues, data is stored within a 'time machine' and can be replayed as needed. Data association occurs as soon as the necessary data becomes available, and it can be recomputed if required. While we have not accounted for latencies at this stage, these capabilities could be used in real-time applications in the future.

A.4.3 *ERASMO*

To properly integrate the final ERASMO solution, we developed tools to enable the launch of all programs across the different PCs involved in ERASMO from our system. This included being able to establish communication between the systems, exchange all necessary information, and record data for the system validation and performance evaluation phases.

Additionally, since we were responsible for the object detection and data association with the map to provide additional localization information, lidar-based traffic sign detectors were developed, and a data association module was built.

This module, based on the one developed during the thesis, associates road markings detected by the Mobileye camera and traffic signs detected by the lidar with the map.

Finally, in addition to the full integration of the solution and the data recording for project validation, a final demonstration was organized. For this, various visualization tools were set up, though not all were used. Furthermore, leveraging the previous developments did in other projects and available at the lab, an autonomous driving demonstration on a test track was successfully carried out. A video introducing the entire project, the developments, and the experimental setup is available on YouTube[1].

[1] https://youtu.be/A6BDxoHM5so?si=pjPia2KUs0DoukQW

Appendix B

Transforming maps and different localization sources in a common working frame

In this thesis, we used HD maps to automatically annotate data and obtain localization information. This was possible because, as mentioned in Chapter 1, these maps contain a set of georeferenced features, including the poles used in our study. As explained in Chapter 2, the coordinates of these features are typically defined in a local Cartesian coordinate system: a ENU frame. Working within such a local frame allows for various transformations of data between the sensor frame and the map frame, which is crucial, as demonstrated throughout this thesis.

However, this implies that any georeferenced point, whether from the map or the vehicle's global position, must be expressed in this frame. When estimating a position using a GNSS receiver, the coordinates obtained are referred to as geodetic coordinates and are based on a reference ellipsoid representing the Earth. For a given point P, they are denoted as (ϕ_P, λ_P, h_P) where ϕ_P is the latitude, λ_P the longitude and h_P the ellipsoidal height.

To convert these coordinates into ENU coordinates, with O as the origin of the ENU frame, we first need to transform the geodetic coordinates into ECEF coordinates, a global Cartesian coordinate system, before converting them into ENU ones. The equations for transforming the geodetic coordinates of a point P into ECEF coordinates are as follows [Subirana et al., 2013]:

$$e = \sqrt{1 - \frac{b^2}{a^2}} \tag{B.1}$$

$$f = 1 - \frac{b}{a} \tag{B.2}$$

$$N(\phi_P) = \frac{a}{\sqrt{1 - e^2 \sin^2(\phi_P)}} \tag{B.3}$$

$$X_P = [N(\phi_P) + h_P] \cos(\phi_P) \cos(\lambda_P) \tag{B.4}$$

$$Y_P = [N(\phi_P) + h_P] \cos(\phi_P) \sin(\lambda_P) \tag{B.5}$$

$$Z_P = \left[(1 - f)^2 N(\phi_P) + h_P\right] \sin(\phi_P) \tag{B.6}$$

where (X_P, Y_P, Z_P) are the ECEF coordinates of point P, a is the semi-major axis (equatorial radius), b is the semi-minor axis (polar radius), e is the eccentricity of the ellipsoid, f is the flattening of the ellipsoid and $N(\phi_P)$ is the prime vertical radius of curvature, i.e., the distance from the surface to the Z-axis along the ellipsoid normal. Then, the equations to convert the ECEF coordinates of point P into ENU coordinates are as follows [Bonnifait et al., 2021]:

$$\begin{bmatrix} x_P \\ y_P \\ z_P \end{bmatrix} = \begin{bmatrix} -\sin(\lambda_O) & \cos(\lambda_O) & 0 \\ -\sin(\phi_O)\cos(\lambda_O) & -\sin(\phi_O)\sin(\lambda_O) & \cos(\phi_O) \\ \cos(\phi_O)\cos(\lambda_O) & \cos(\phi_O)\sin(\lambda_O) & \sin(\phi_O) \end{bmatrix} \begin{bmatrix} X_P - X_O \\ Y_P - Y_O \\ Z_P - Z_O \end{bmatrix} \tag{B.7}$$

where (ϕ_O, λ_O, h_O) are the geodetic coordinates and (X_O, Y_O, Z_O) are the ECEF coordinates of point O, the origin of the ENU frame.

It appears that converting from geodetic coordinates to ENU ones can be straightforward. However, one must consider how geodetic coordinates are obtained and whether there is a single, universal definition for them. If not, it is crucial to ensure that the same definition is used for both the ENU origin and the points being expressed in ENU frame. Figure B.1 illustrates trajectories composed of points with two different geodetic coordinates obtained from receivers configured differently. Additionally, the HD map features are also displayed in the figure using their geodetic coordinates.

Here, the observed GNSS trajectories are from solutions with centimeter-level accuracy: There is a noticeable shift between the two trajectories. The blue trajectory appears to incorrectly place the vehicle on its lane, suggesting that it crossed a lane border corresponding to a sidewalk. The green trajectory seems to correspond to the true trajectory followed by the vehicle. This is due to the fact that the geodetic coordinates were obtained using different settings between the receiver that provided the blue trajectory and the mapping solution used to create the HD map.

Indeed, even though the trajectories and the map were obtained using highly accurate GNSS receivers, the geodetic coordinates rely on different reference systems. For a given receiver, the geodetic coordinates are determined relative to a specific coordinate system defined by a geodetic datum.

A geodetic datum is a model that defines the shape, size, a reference point and the orientation of the Earth or a local region. It helps measure and compare positions on the Earth's surface. It consists of a reference ellipsoid (notably used for ECEF conversion), and a coordinate system. The datum's origin and orientation are determined to align the ellipsoid with the Earth's center or a specific local point. Geodetic datums can be global, such as WGS84 (World Geodetic System 1984) used by all GNSS systems, or regional, like the RGF93 (Réseau Géodésique Français) in France. Other local reference systems exist, such as ETRS89 used across Europe. They optimize positioning accuracy for different scales. WGS84 is ideal for global applications due to its widespread use in GNSS and international systems and the capacity to provide a consistent reference frame across the world.

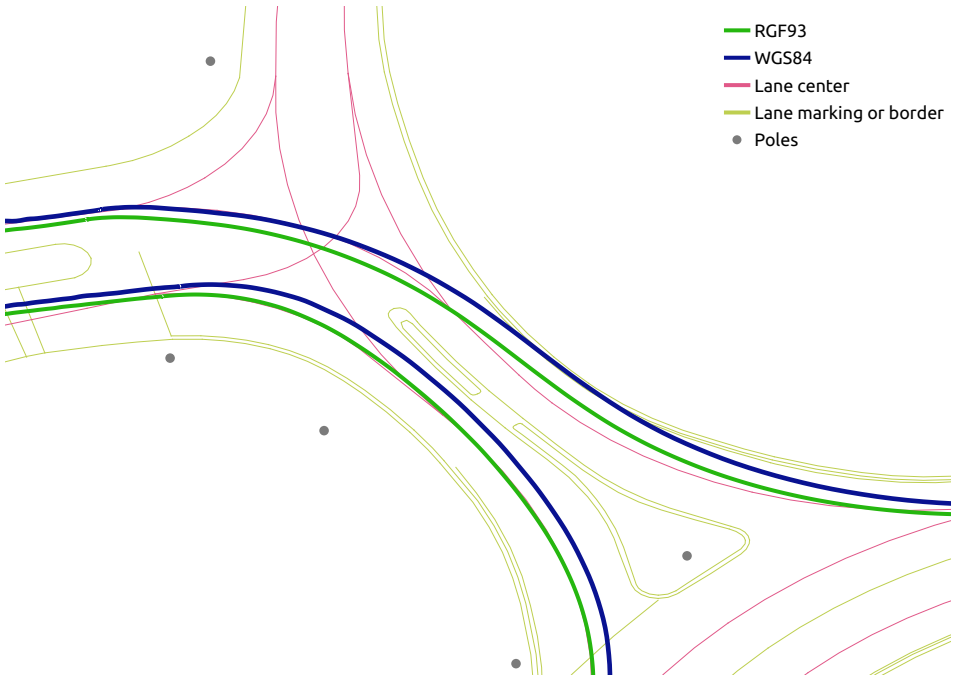


Figure B.1: *Comparison of two trajectories obtained from different GNSS receivers. The geodetic coordinates (latitude and longitude) of both the map features and trajectories are displayed. The green trajectory uses the same geodetic datum as the map, while the blue trajectory uses a different geodetic datum, resulting in the observed variable shift.*

RGF93 offers enhanced accuracy for regional applications within France as local mapping. Typically, in the figure the green trajectory and the map use the RGF93 datum while the blue trajectory uses the WGS84 datum.

Generally, the geodetic datum is defined, updated and maintained using a network of reference stations on the Earth. WGS84 uses a network of stations around the world to keep the geodetic reference system accurate. These stations are strategically placed around the world to ensure general coverage. The data from these stations are regularly incorporated to account for various changes, such as tectonic plate movements, and to update the datum. When a datum is updated using data collected up to a certain epoch, this process is referred to as a 'realization'. The same principle is applied for RGF93 where stations across France collect geodetic data for new realizations.

Please note, without delving into the details, that the aim of the RGF93 system is to provide a more accurate local reference system than a global one. The coordinates in RGF93 change very little over time because RGF93 accounts for the drift of the French territory due to tectonic plate movements. In contrast, global systems show changing coordinates for points in France over time. This stability makes RGF93 highly favored for applications like mapping.

Therefore, for ENU conversion, it is not enough to simply know the geodetic coordinates of a point P and the origin O. One must also know the geodetic datum and the date when the coordinates were obtained to identify the correct

reference system to use in the transformation. To convert P coordinates to ENU, it is essential to also know the geodetic coordinates of O (the origin of the ENU frame) within the same datum and the same realization. Thus, O must be accurately measured using a highly accurate GNSS receiver. In our case, we use the GNSS receiver from our base station, as explained in Appendix A.

If we omit the reference system realizations for simplicity, the green trajectory in Figure B.1 consists of points in the RGF93 system, while the other trajectory uses points in the WGS84 system. The map itself is in RGF93. To transform all these elements into the same ENU frame, it is necessary to know the parameters for both RGF93 and WGS84, as well as the coordinates of O in both datums. The correct parameters must be applied to each data source to align everything within the same local frame.

Let us consider a practical example. Suppose the map features are georeferenced in RGF93 coordinates. To express them in the ENU frame, the geodetic coordinates of the origin must also be in RGF93, and the ellipsoid parameters must correspond to those used in RGF93 (the GRS80 ellipsoid in this case). If a position provided by a GNSS receiver is in WGS84, to convert it into ENU coordinates, the geodetic coordinates of the origin must be in WGS84, and the corresponding ellipsoid parameters (the WGS84 ellipsoid in this case) must be used.

This illustrates the advantage of working with a local reference frame. First, it allows the transformation of geodetic coordinates into Cartesian coordinates without requiring a projection, thereby avoiding the introduction of errors. Second, when different datums are used, each applies its own ENU transformation using the correct parameters for the respective datums. This is made possible by using the proper geodetic coordinates and datum for the origin of the ENU reference frame.

Appendix C

Empirical function for ground search area definition

In Section 2.3, we use lidar data to enhance map-based annotations for images. As illustrated in Figure 2.4, the 2D hypothesis may lead to projections that do not align with the base of the pole in the image. Therefore, an estimation of ground elevation is required to adjust the annotation's height accurately.

To address this issue, we propose correcting the Z-coordinate of the annotation within the lidar frame before projecting it into the camera frame and subsequently onto the image. This approach is depicted in Figure C.1.

The process begins by identifying ground points in the lidar point cloud using a ground segmentation algorithm, shown as blue dots. For each map element, represented as orange crosses, these ground points are searched within a red-highlighted zone. As the distance from the sensor increases, the search area must expand to account for the increasingly sparse point cloud. However, an excessively large zone, whether near or far from the sensor, risks incorporating ground points with elevations that differ significantly from the base of the pole. Thus, the goal is to define the smallest possible search zone that still captures enough ground points for an accurate elevation estimate. Note that if no ground points are found within the search area, the corresponding map element is discarded and not annotated in the image to prevent incorrect annotations.

Using this method, the map elements are corrected by computing the average Z-value of the ground points within the identified zone as defined in Eq. (2.7), resulting in green crosses that represent the adjusted positions of the map elements.

A 2D view of the problem is shown in Figure C.2. To define the search area for each map element, we make the assumption that the empirical function provides the search area as if the ground were perfectly orthogonal to the vehicle. If this assumption does not hold true for the area around the target map element, the search zone may become slightly oversized, potentially including points that do not correspond to the elevation of the pole's base. Nevertheless, this assumption is necessary and simplifies the process of defining the search zone's size.

First, we observe that certain areas around the vehicle are not scanned, as indicated by the missing coverage of the lidar beams shown in blue. If a pole is located within one of these blind spots, it must be disregarded because its base is not visible in the lidar data.

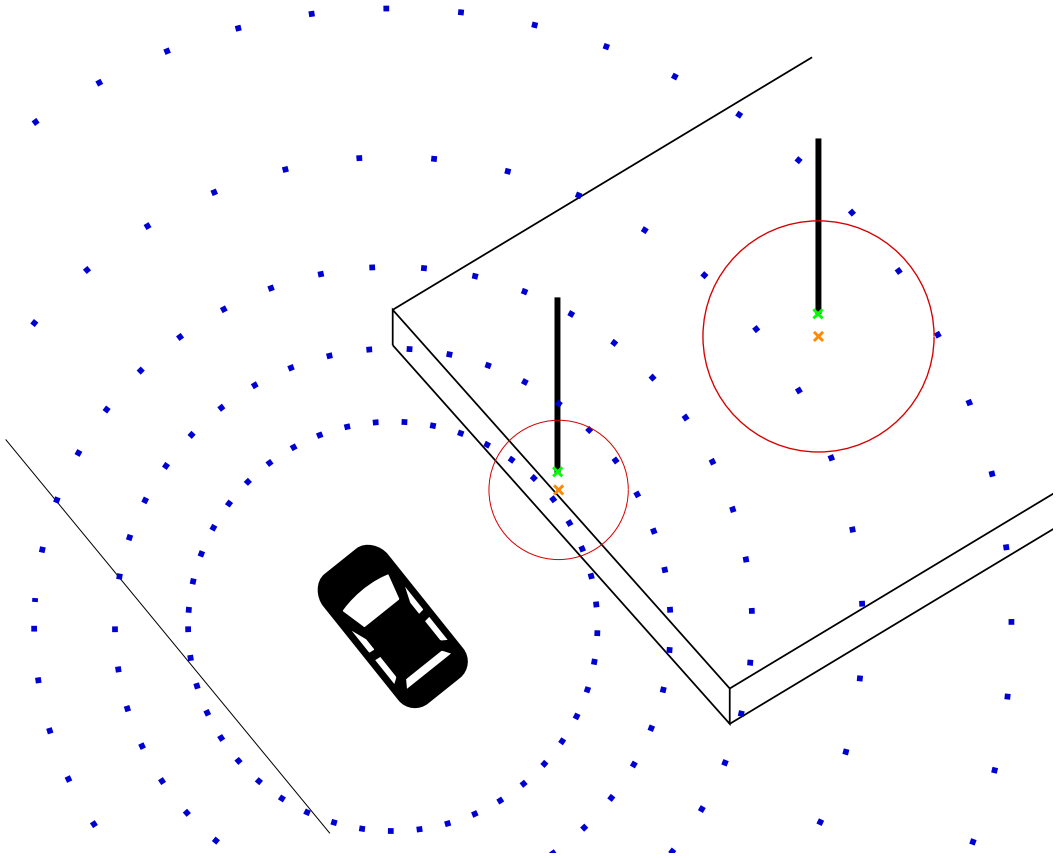


Figure C.1: *Search zones for ground points used to correct the height of each mapped pole base. Ground points are shown in blue, initial map data in orange, and corrected positions in green. The search zones, displayed in red, expand as the distance between the pole and the sensor increases, due to the growing sparsity of the data.*

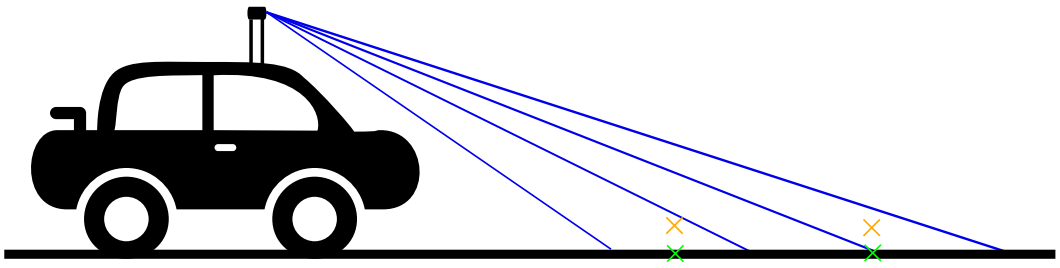


Figure C.2: *Ground projections (green) of two map points (orange): one illustrates a scenario where no lidar ring scans the ground at the base of the pole, while the other depicts a case where the lidar ring covers the corresponding area*

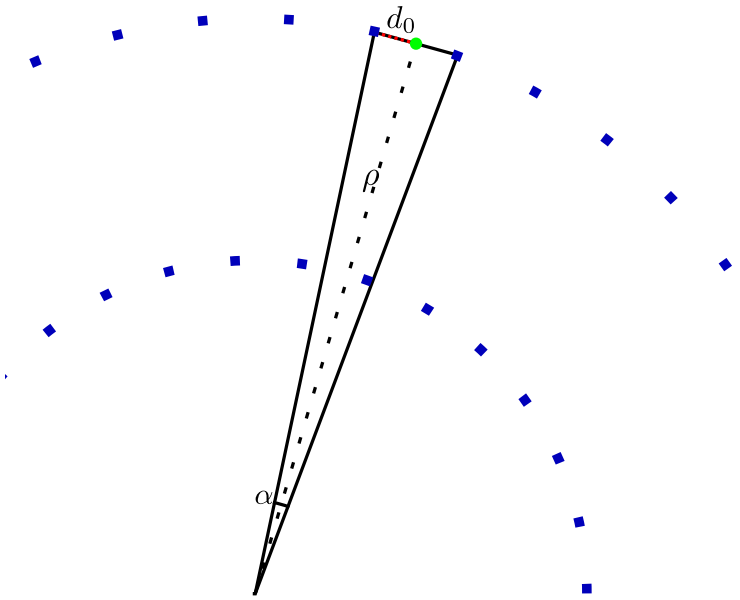


Figure C.3: *Maximum distance d_0 along a ring between the map point (green) and the nearest lidar point (blue), computed using the sensor's horizontal resolution α and the 2D distance to the map feature ρ, under the assumption of orthogonal ground hypotheses and with the map point positioned exactly on the ring.*

Next, two possible scenarios can arise. In the first scenario, the map element (represented in orange based on the map data and in green on the orthogonal plane) lies directly on a theoretical lidar ring. In this case, the distance between the map element and the nearest point on the ring is calculated to define the search area. In the second scenario, the map element is located between two theoretical rings. In this case, the minimum distance between the map element and the closest ring is determined to establish the search area.

As previously mentioned, it is critical to avoid overly large search zones. Therefore, the maximum 2D distance between a ground point and the map element is limited to 3 meters.

C.1 Minimum Worst-Case Distance on a Theoretical Lidar Ring Between a Lidar Point and a Map Element

First, let us focus on the case where the map element lies exactly on a theoretical ring of the point cloud. As illustrated in Figure C.3, let ρ denote the 2D distance between the map element and the sensor (ignoring any height difference), which corresponds to the radius of the ring. Let α represent the horizontal resolution of the sensor. As shown in the figure, in the worst-case scenario, the map element is equidistant from the two nearest lidar points on the ring of interest.

The minimum distance d_0 required to ensure a lidar point is within the search area can be determined using the isosceles triangle formed by one of the nearest

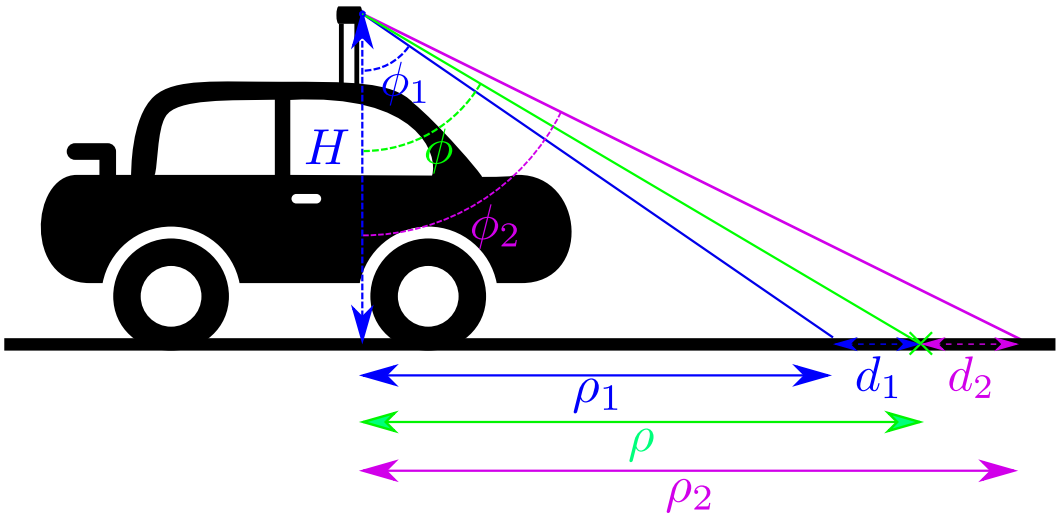


Figure C.4: *Estimation of the minimum distance between the map element (projected onto the orthogonal plane in green) and the closest theoretical rings (in blue and purple). To identify the closest rings, the incidence angle ϕ for the map element is estimated using the sensor height H and the distance to the map element ρ. The angles ϕ_1 and ϕ_2 correspond to the incidence angles of the closest rings. The distances to the sensor for both rings, ρ_1 and ρ_2, are then calculated, allowing for the estimation of the two distances, d_1 and d_2, between the map element and the rings. The minimum distance is kept.*

lidar points, the position of the map element (in green), and the position of the sensor. This distance is given by the following equation:

$$d_0 = \rho \sin\left(\frac{\alpha}{2}\right) \tag{C.1}$$

To account for more points in the search area and also consider the distance variations in a real-world scenario, a multiplicative factor can be applied either to the distance d_0 or to the incidence angle α to simulate a more sparse sensor.

C.2 Minimum distance between the map element and a theoretical point on the nearest lidar ring

Now, let us consider the case where the map element lies between two theoretical rings of the point cloud, as illustrated in Figure C.4.

To precisely identify these two rings, represented by the blue and purple beams, it is first necessary to estimate a theoretical incidence angle ϕ for the map element, represented by the green beam. This angle can be calculated using the sensor height H and the distance ρ to the map element:

$$\phi = \tan^{-1}\left(\frac{\rho}{H}\right) \tag{C.2}$$

This estimation also helps determine whether the map element is located exactly on a ring or in an area outside the sensor's visibility.

Once ϕ is calculated, the incidence angles ϕ_1 and ϕ_2 corresponding to the two closest theoretical rings can be derived using the sensor's specifications. From these angles, the 2D distances ρ_1 and ρ_2 between the sensor and the two rings can be computed using a similar equation:

$$\rho_i = \tan(\phi_i) H \tag{C.3}$$

for $i = \{1, 2\}$

With these values, the distances d_1 and d_2 can be determined, where d_1 is the distance between the map element and the preceding ring, and d_2 is the distance between the map element and the following ring.

However, at this stage, there is no guarantee that these distances correspond to the minimum distance to a point on either ring. This is due to the horizontal resolution of the sensor, which may result in no lidar points being present within these distances.

To address this, we propose slightly increasing the values of d_1 and d_2 by taking into account the theoretical spacing between points on each ring:

$$D_1 = \sqrt{d_1^2 + \left(\rho_1 \sin\left(\frac{\alpha}{2}\right)\right)^2} \tag{C.4}$$

$$D_2 = \sqrt{d_2^2 + \left(\rho_2 \sin\left(\frac{\alpha}{2}\right)\right)^2} \tag{C.5}$$

However, as shown in Figure C.5 for the first ring, the distance may be underestimated due to an angular error δ between the tangent to the circle, where the additional distance is applied in our formula, and the segment representing the true minimum distance. This discrepancy is considered negligible or effectively mitigated by applying a multiplicative factor to the added distance or to the angle α, as discussed earlier.

This adjustment ensures the presence of at least one lidar point within the search area, while accounting for the sensor's point density.

To ensure a sufficient number of lidar points within the search area, we select the larger value between D_1 and D_2 as the search radius. However, this radius is capped at 3 meters to prevent the search from extending too far, which could incorporate elevation variations that compromise accuracy. The primary objective of this approach is to minimize the search area, particularly for nearby poles, to achieve the most accurate estimation of the pole base position.

It is important to note that in the results presented in this thesis, the multiplicative factors were incorrectly applied, leading to search areas larger than initially intended. Despite this error, the annotation performance remained unaffected. This outcome suggests that the function may not be as critical for ground point detection as initially assumed, or that further improvements could be achieved by refining the definition of the ground point search area.

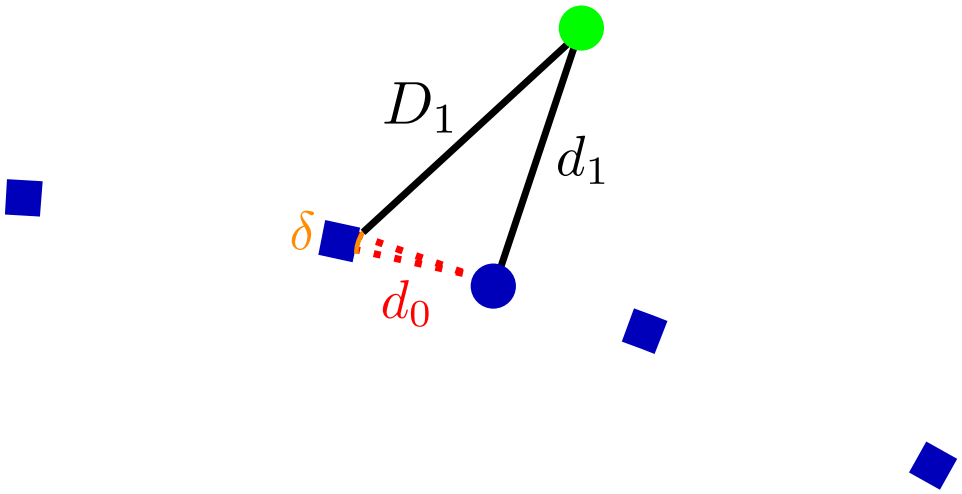


Figure C.5: *Angular error δ introduced when defining the distance D_1 using d_1 and d_0, caused by the use of the tangent to the circle for applying d_0. While this error appears negligible at close range, it increases with data sparsity and, consequently, with the distance from the sensor.*

Appendix D

Study of impact of annotation errors on pole base detection performance: all curves

Contents

In Chapter 3, we conducted an initial study on the impact of annotation errors on detection performance. To achieve this, we defined models to simulate diverse errors: spawns, drops and positioning errors. To simplify the study, we focused on the evolution of the Average Precision (AP) as a function of the error level, whether it be the percentage of occurrences for spawns and drops or the maximum error in the case of simulated annotation noise. This approach allowed us to summarize a large set of training sessions and PR curves into simple values. Below are all the PR curves leading to similar conclusions. Some PR curves reveal a significant drop in performance, which may result from learning issues or error generation with a particularly pronounced impact compared to other generations used in the study. For a better analysis, additional trains with various random generations of errors should be carried out to account for uncertainty. Although this is only a preliminary study, it offers valuable initial insights into the effects of annotation errors, as summarized in the corresponding chapter.

D.1 Detectors overall performance under spawn influence

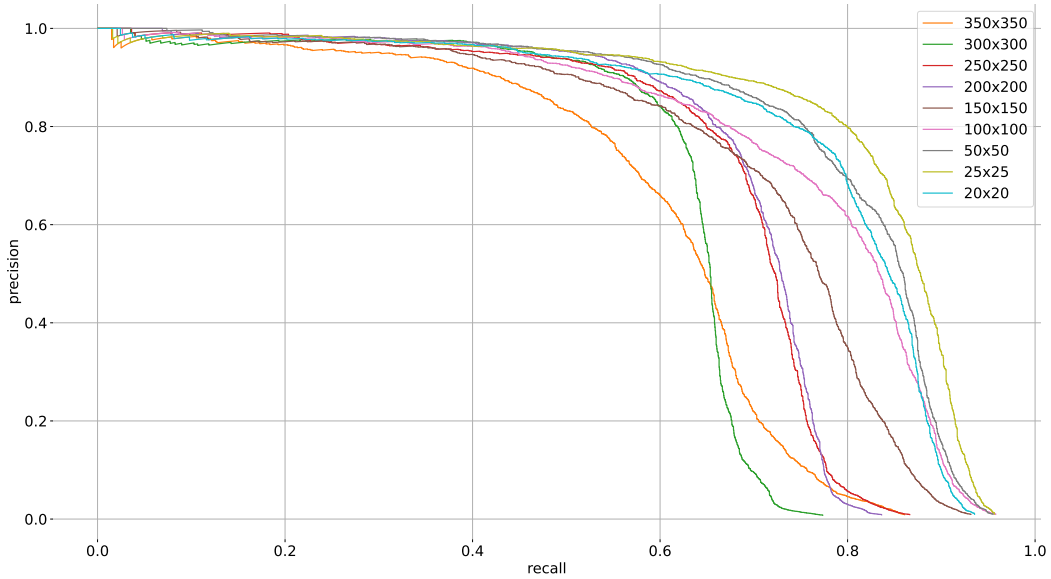


Figure D.1: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated spawns such that* $\gamma_{sp} = 0.1$.

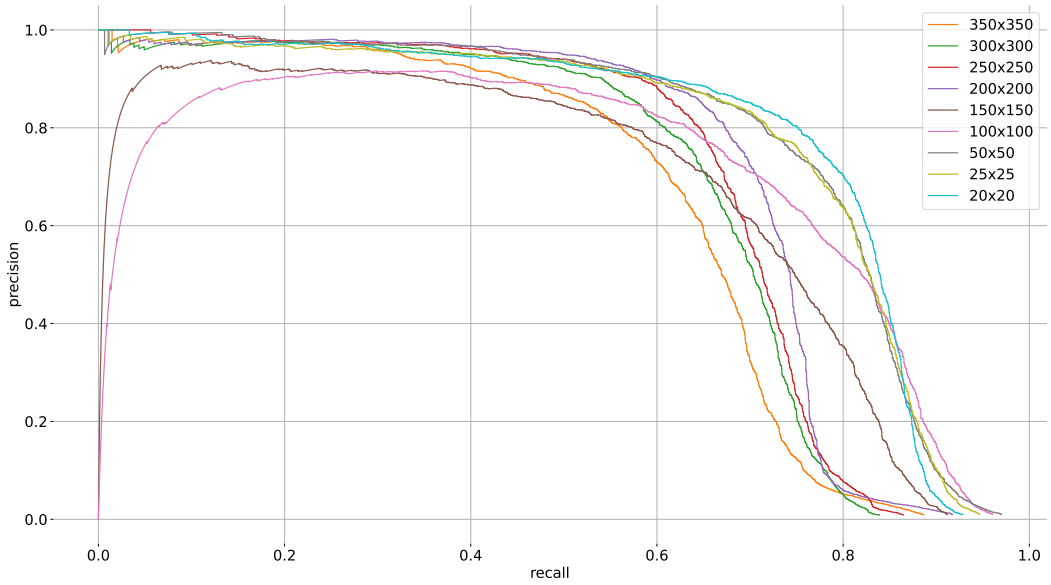


Figure D.2: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated spawns such that* $\gamma_{sp} = 0.2$.

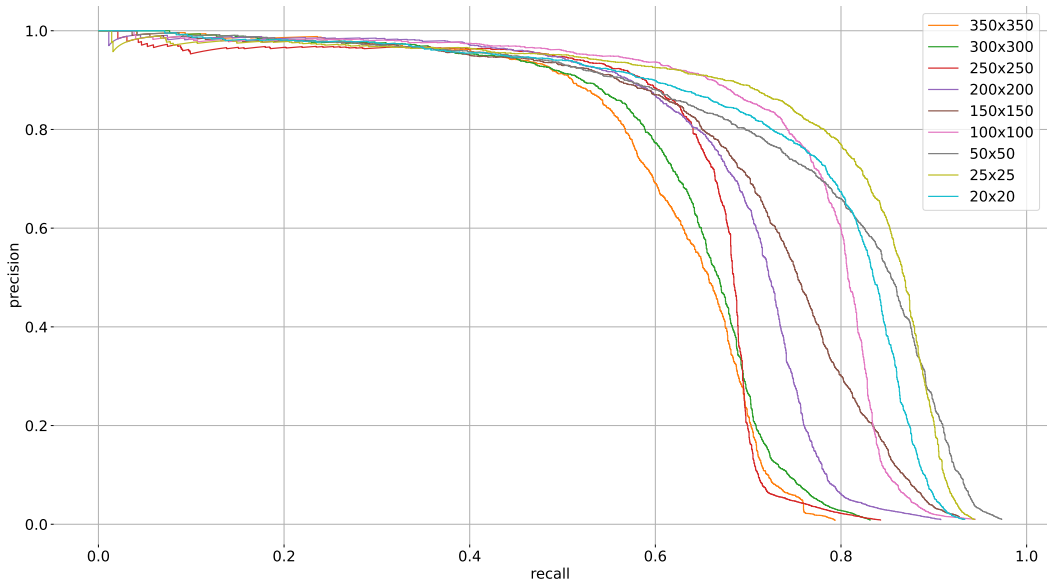


Figure D.3: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated spawns such that* $\gamma_{sp} = 0.3$.

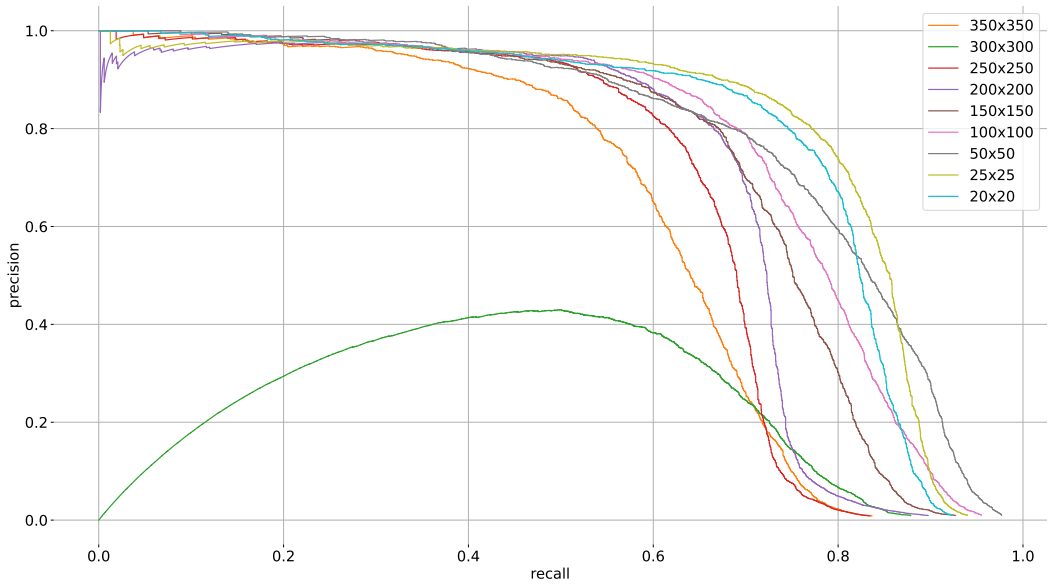


Figure D.4: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated spawns such that* $\gamma_{sp} = 0.4$.

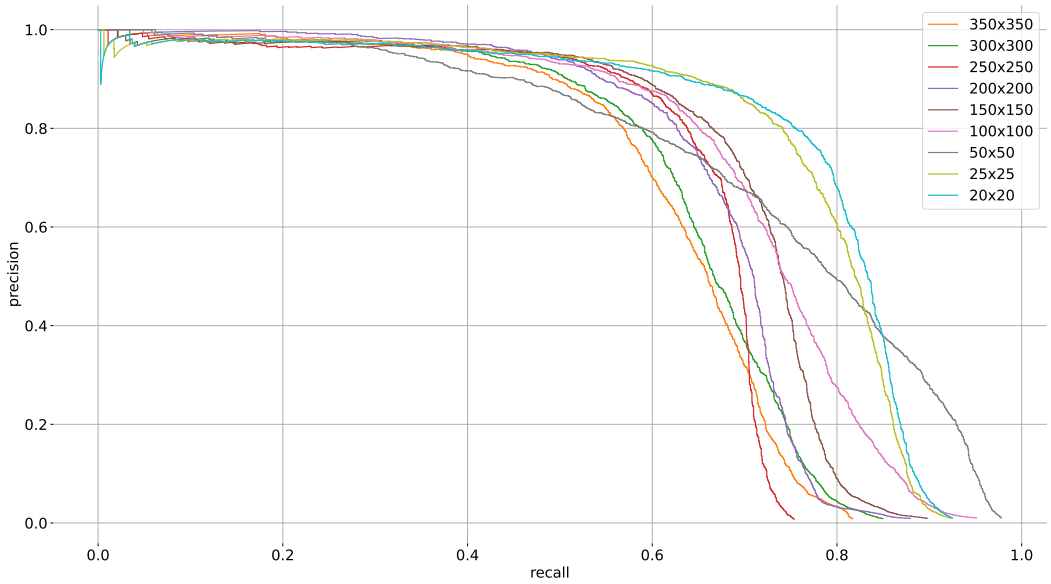
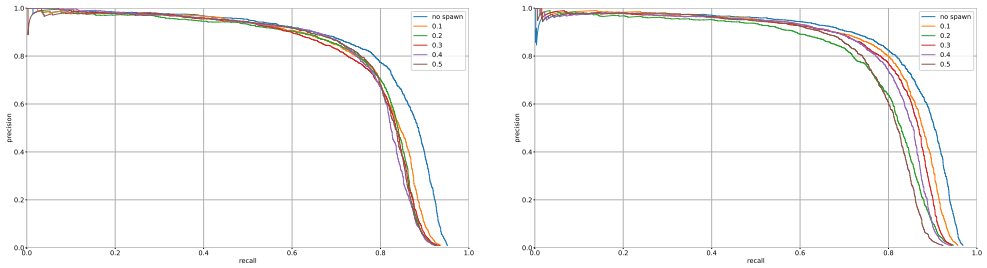
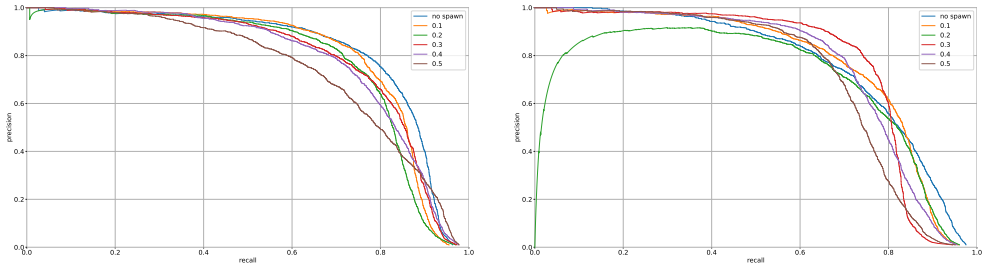


Figure D.5: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated spawns such that* $\gamma_{sp} = 0.5$.

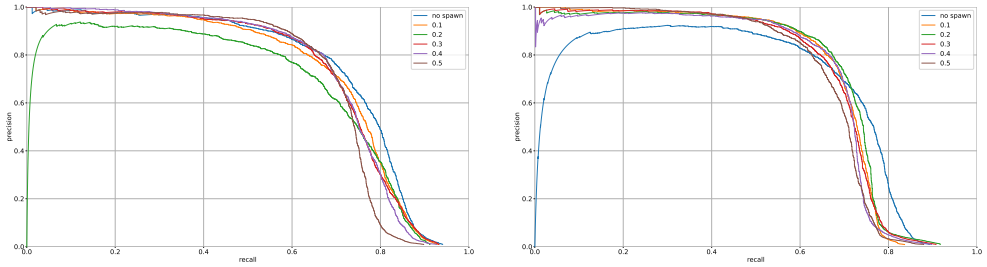
D.2 DETECTORS OVERALL PERFORMANCE UNDER DROP INFLUENCE



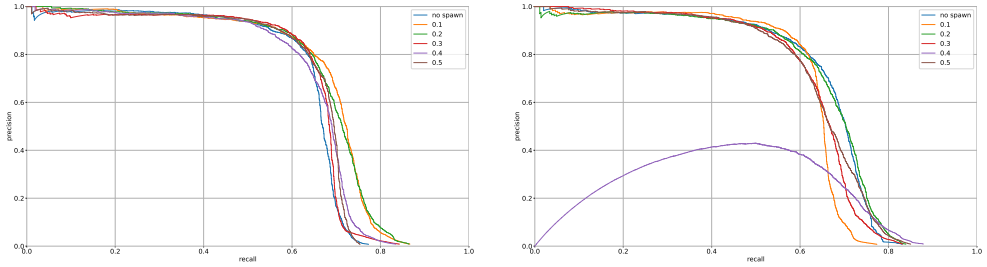
(a) *20x20* ***(b)*** *25x25*



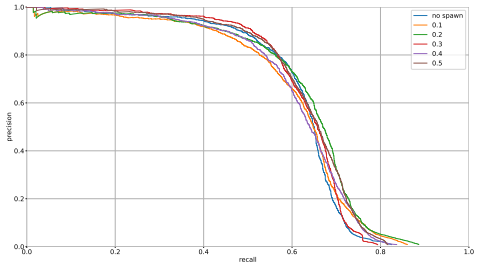
(c) *50x50* ***(d)*** *100x100*



(e) *150x150* ***(f)*** *200x200*



(g) *250x250* ***(h)*** *300x300*



(i) *350x350*

Figure D.6: *Precision-Recall curves obtained during spawn influence study after 300 epochs of training. For each box size, all PR curves obtained such that* $\gamma_{\text{sp}} \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ *are summarized in a same figure.*

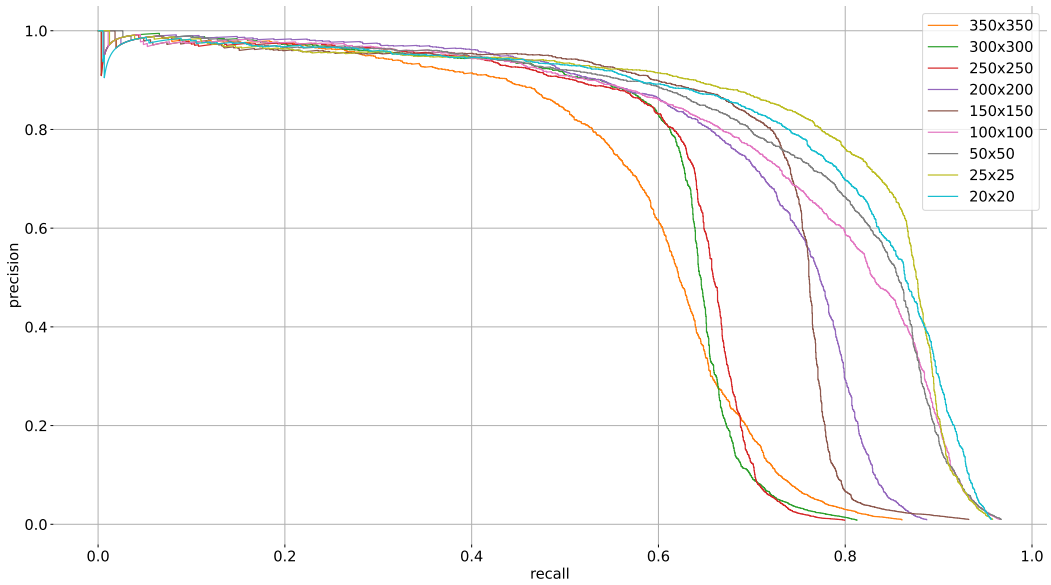


Figure D.7: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated drops such that* $\gamma_d = 0.1$.

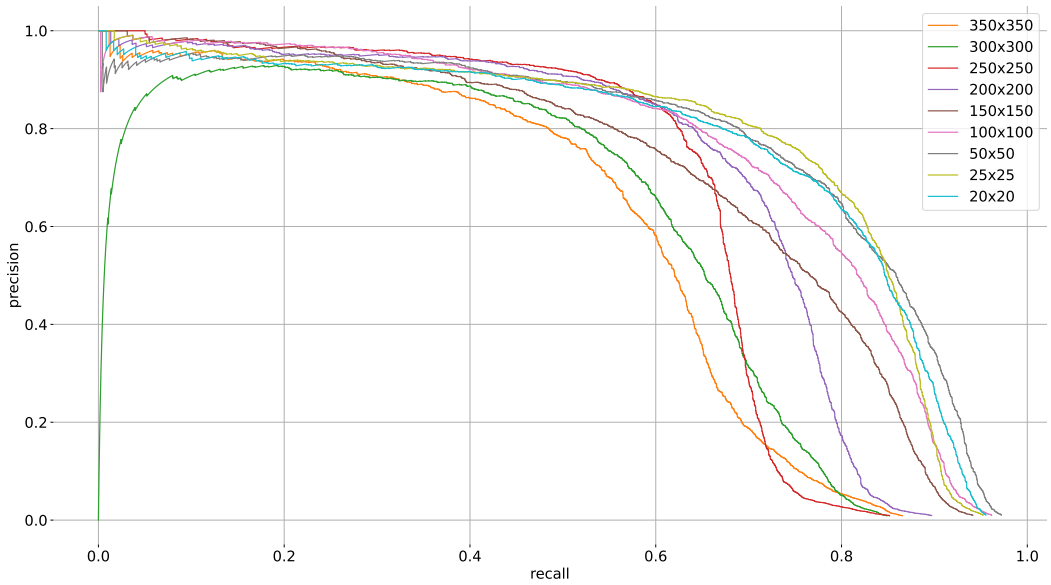


Figure D.8: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated drops such that* $\gamma_d = 0.2$.

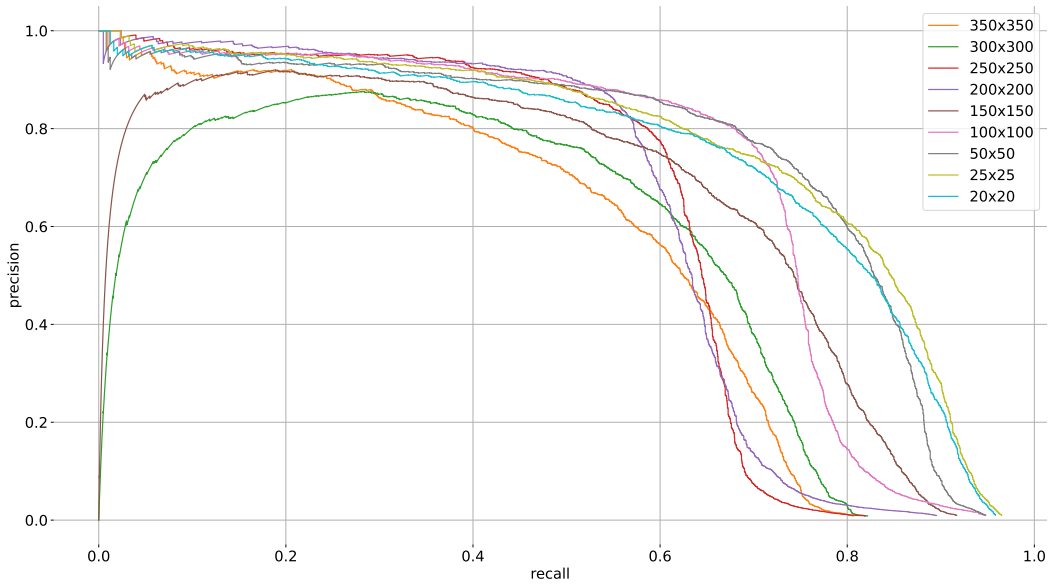


Figure D.9: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated drops such that* $\gamma_d = 0.3$.

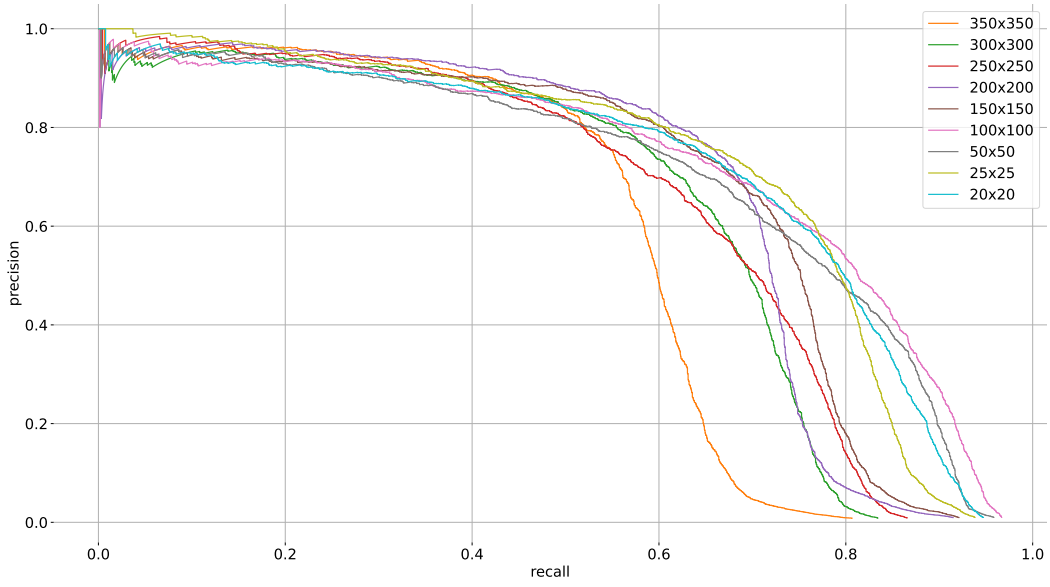


Figure D.10: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated drops such that* $\gamma_d = 0.4$.

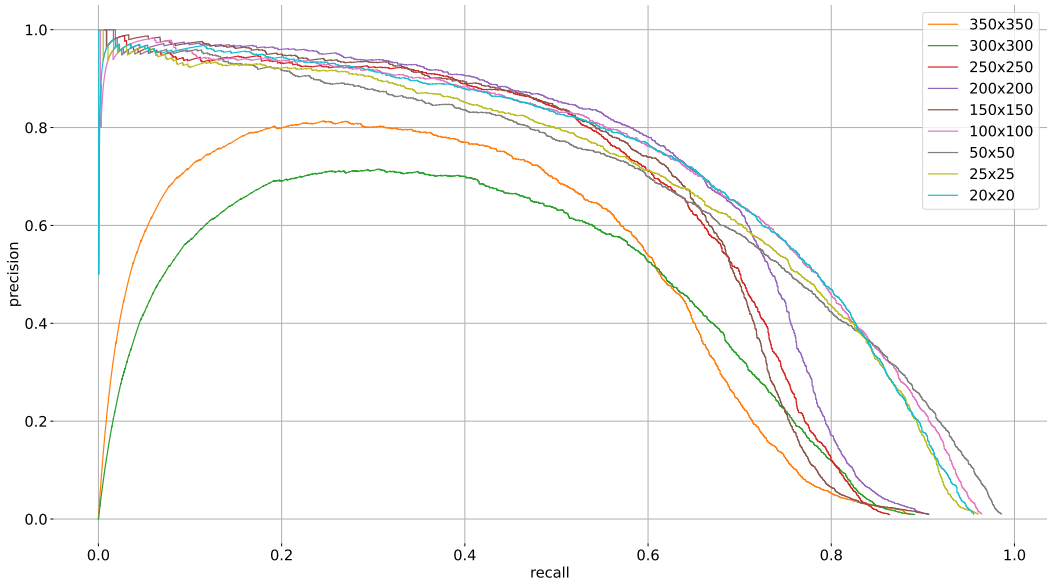
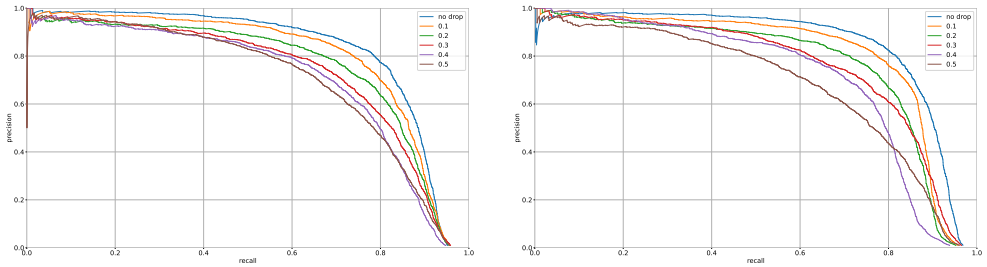


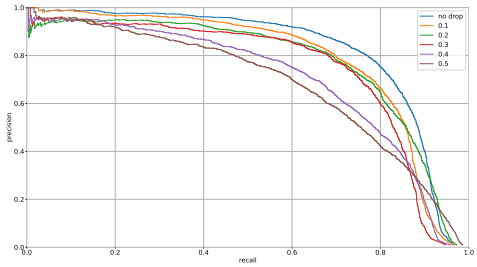
Figure D.11: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated drops such that* $\gamma_d = 0.5$.

D.3 DETECTORS OVERALL PERFORMANCE UNDER NOISE INFLUENCE

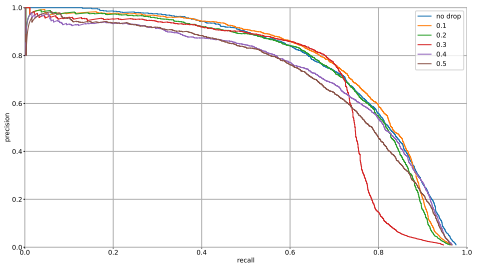


(a) 20x20

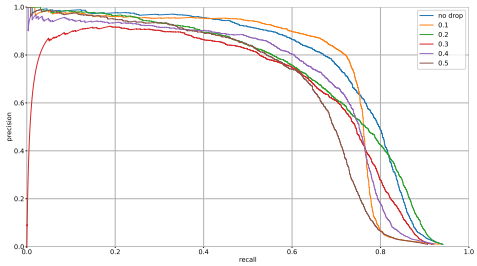
(b) 25x25



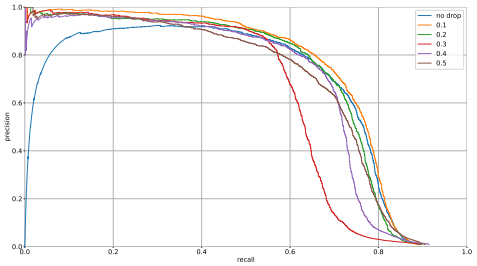
(c) 50x50



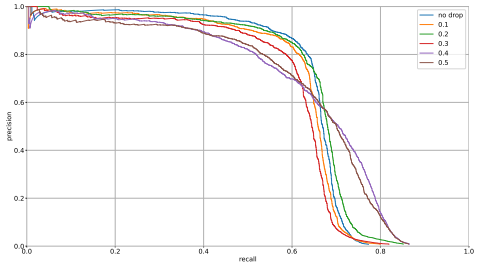
(d) 100x100



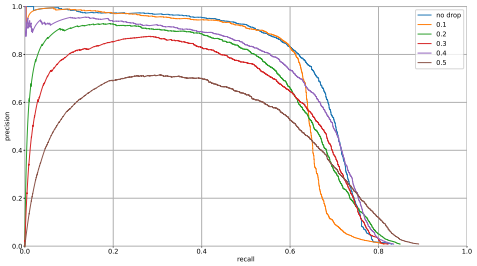
(e) 150x150



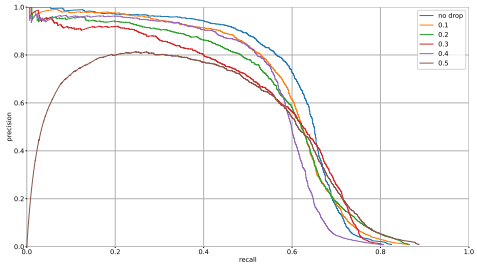
(f) 200x200



(g) 250x250



(h) 300x300



(i) 350x350

Figure D.12: *Precision-Recall curves obtained during drop influence study after 300 epochs of training. For each box size, all PR curves obtained such that* $\gamma_d \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ *are summarized in a same figure.*

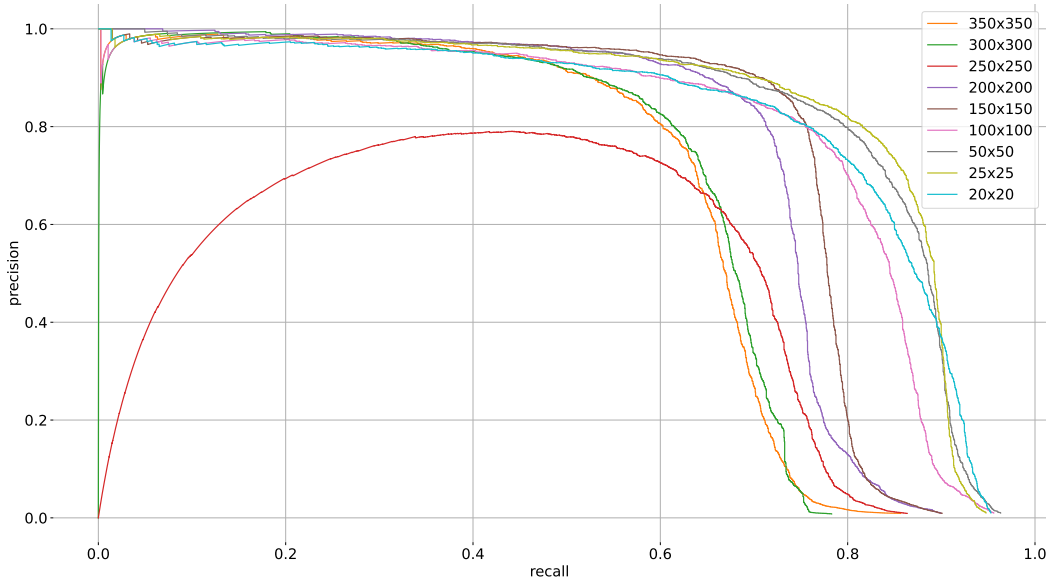


Figure D.13: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated positioning noise such that* $\epsilon = 2$.

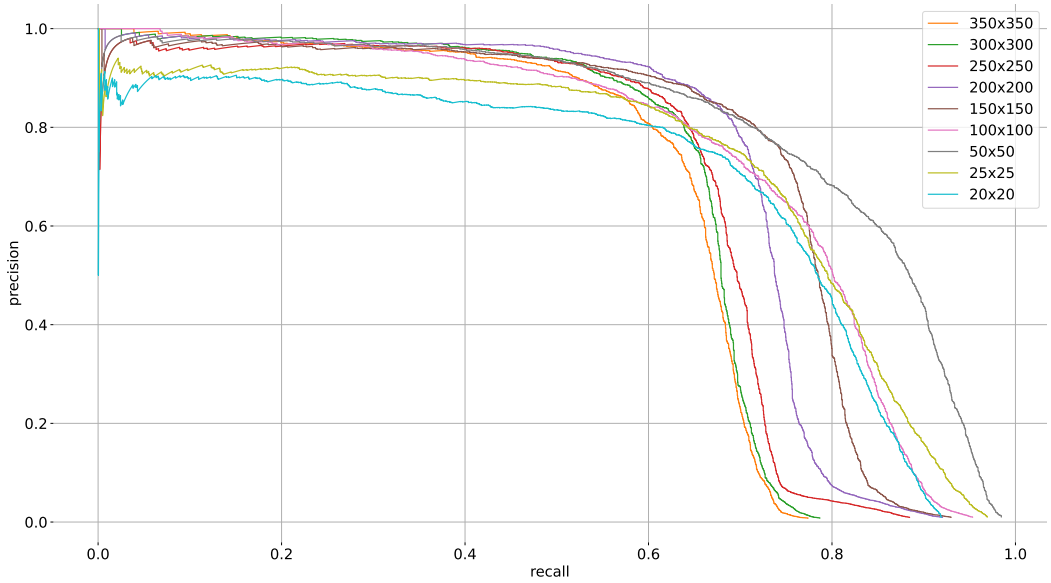


Figure D.14: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated positioning noise such that* $\epsilon = 5$.

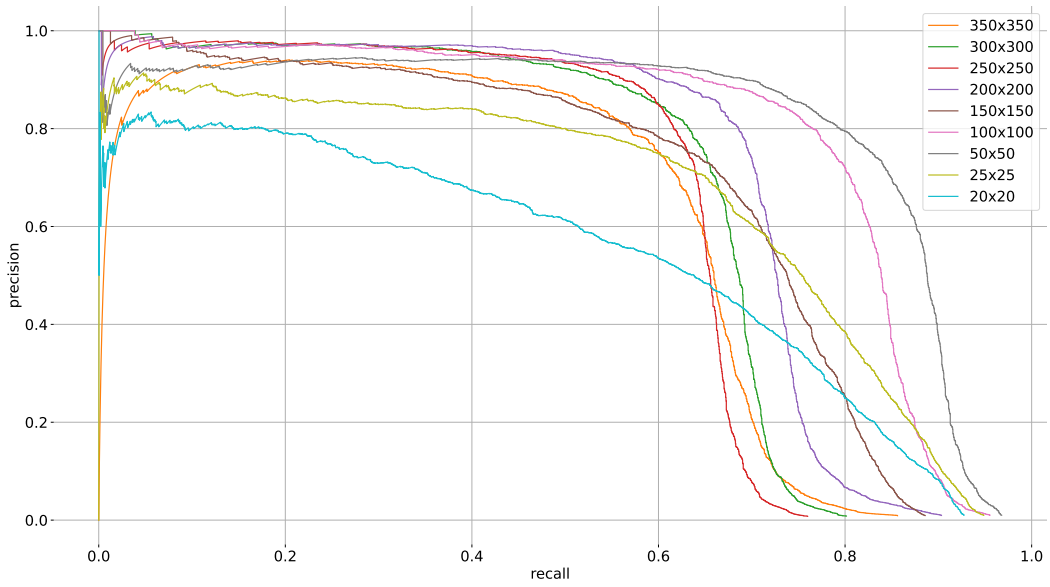


Figure D.15: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated positioning noise such that* $\epsilon = 7$.

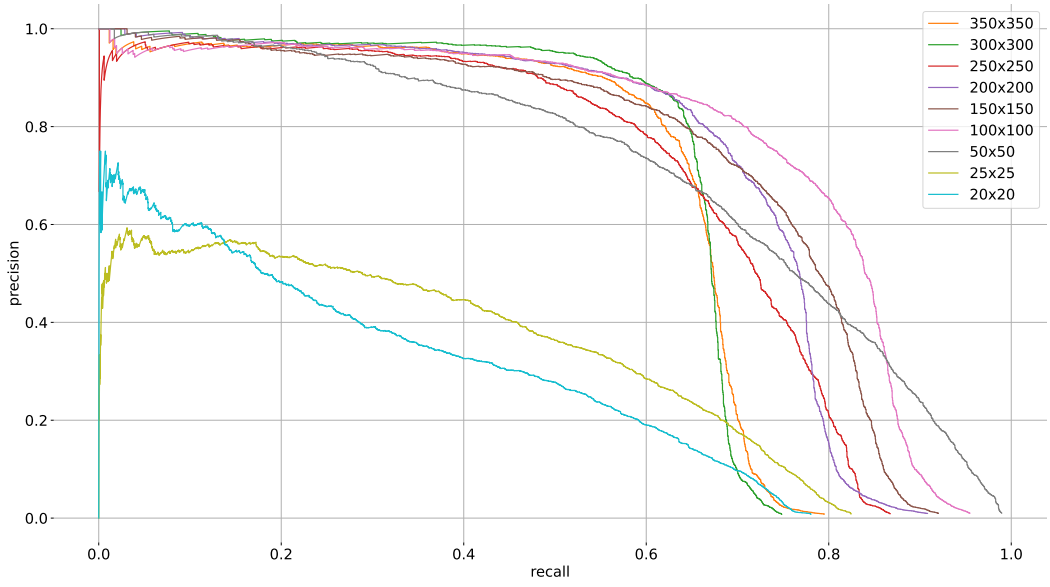


Figure D.16: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated positioning noise such that* $\epsilon = 10$.

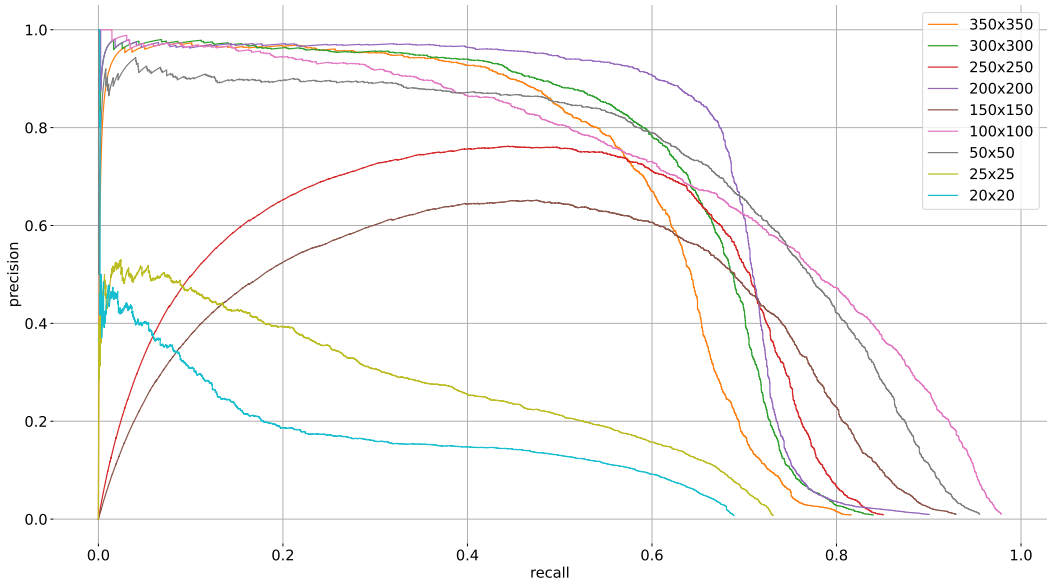
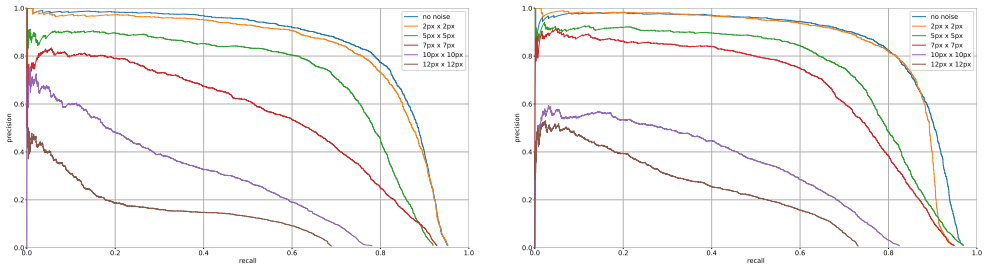
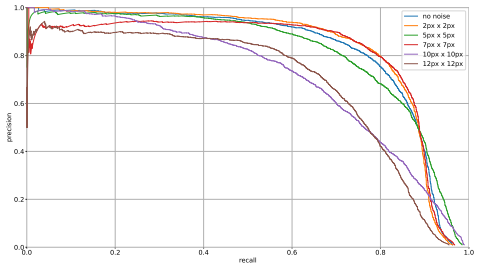


Figure D.17: *Precision-Recall curves obtained after 300 epochs of training with different box sizes using manually annotated images with simulated positioning noise such that* $\epsilon = 12$.

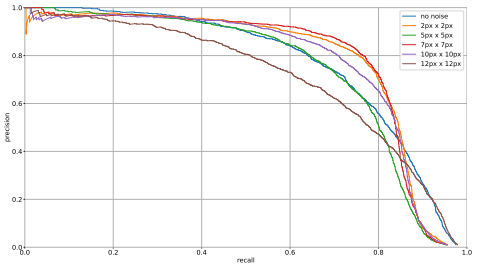


(a) 20x20

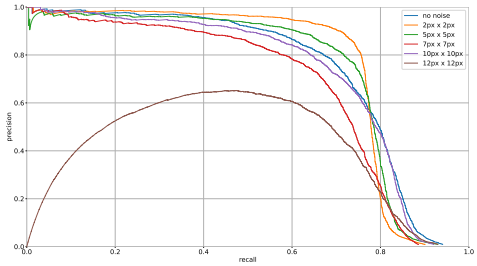
(b) 25x25



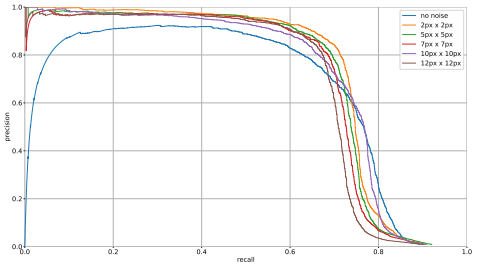
(c) 50x50



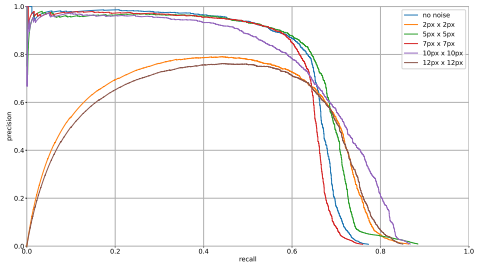
(d) 100x100



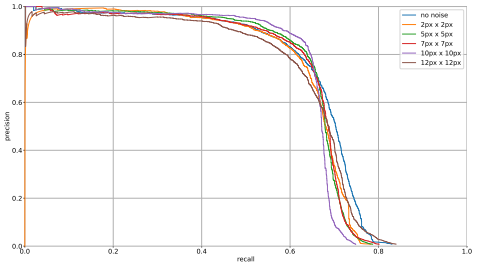
(e) 150x150



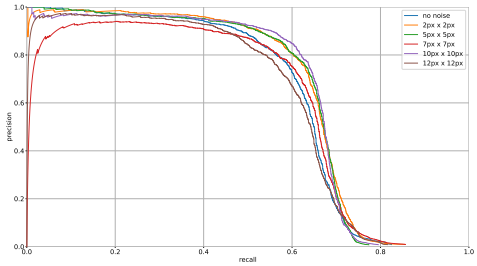
(f) 200x200



(g) 250x250



(h) 300x300



(i) 350x350

Figure D.18: *Precision-Recall curves obtained during noise influence study after 300 epochs of training. For each box size, all PR curves obtained such that* $\epsilon \in \{0, 2, 5, 7, 10, 12\}$ *are summarized in a same figure.*

Detectors overall performance under noise influence

Bibliography

Aeberhard, M., S. Rauch, M. Bahram, G. Tanzmeister, J. Thomas, Y. Pilat, F. Homm, W. Huber, and N. Kaempchen (2015). "Experience, Results and Lessons Learned from Automated Driving on Germany's Highways". In: *IEEE Intelligent Transportation Systems Magazine* 7.1, pp. 42–57. DOI: 10.1109/MITS.2014.2360306 (cited on p. 14).

Al Hage, J., P. Xu, and P. Bonnifait (2019). "High Integrity Localization With Multi-Lane Camera Measurements". In: *IEEE Intelligent Vehicles Symposium*, pp. 1232–1238. DOI: 10.1109/IVS.2019.8813988 (cited on p. 99).

Alaba, S. Y. and J. E. Ball (Dec. 2022). "A Survey on Deep-Learning-Based LiDAR 3D Object Detection for Autonomous Driving". en. In: *Sensors* 22.24, p. 9577. ISSN: 1424-8220. DOI: 10.3390/s22249577 (cited on p. 126).

Alexiou, E., X. Zhou, I. Viola, and P. Cesar (Nov. 2022). *PointPCA: Point Cloud Objective Quality Assessment Using PCA-Based Descriptors*. en. arXiv:2111.12663 [cs] (cited on p. 133).

Arana, G. D., O. A. Hafez, M. Joerger, and M. Spenko (Nov. 2020). "Integrity monitoring for Kalman filter-based localization". en. In: *The International Journal of Robotics Research* 39.13, pp. 1503–1524. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364920960517 (cited on p. 28).

Aynaud, C., C. Bernay-Angeletti, R. Aufrere, L. Lequievre, C. Debain, and R. Chapuis (Sept. 2017). "Real-Time Multisensor Vehicle Localization: A Geographical Information System Based Approach". en. In: *IEEE Robotics & Automation Magazine* 24.3, pp. 65–74. ISSN: 1070-9932. DOI: 10.1109/MRA.2017.2669399 (cited on p. 23).

Bahlmann, C., Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler (2005). "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information". en. In: *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* Las Vegas, NV, USA: IEEE, pp. 255–260. ISBN: 978-0-7803-8961-8. DOI: 10.1109/IVS.2005.1505111 (cited on p. 25).

Bailey, T. (2002). "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments". PhD thesis. The University of Sydney, p. 212 (cited on p. 96).

Bao, Z., S. Hossain, H. Lang, and X. Lin (2022). *High-Definition Map Generation Technologies For Autonomous Driving*. arXiv: 2206.05400 [cs.RO] (cited on pp. 16, 20).

Bar-Shalom, Y. (1987). *Tracking and Data Association*. USA: Academic Press Professional, Inc. ISBN: 0120797607 (cited on p. 93).

Barbosa, B., N. Bhatt, A. Khajepour, and E. Hashemi (Jan. 2021). "Soft Constrained Autonomous Vehicle Navigation using Gaussian Processes and Instance Segmentation" (cited on p. 61).

Bauer, S., Y. Alkhorshid, and G. Wanielik (Nov. 2016). "Using High-Definition maps for precise urban vehicle localization". en. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Rio de Janeiro, Brazil: IEEE, pp. 492–497. ISBN: 978-1-5090-1889-5. DOI: 10.1109/ITSC.2016.7795600 (cited on p. 22).

Bekris, K.E., M. Click, and E.E. Kavraki (2006). "Evaluation of algorithms for bearing-only SLAM". en. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* Orlando, FL: IEEE, pp. 1937–1943. ISBN: 978-0-7803-9505-3. DOI: 10.1109/ROBOT.2006.1641989 (cited on pp. 19, 100).

Betke, M. and L. Gurvits (1997). "Mobile robot localization using landmarks". In: *IEEE Transactions on Robotics and Automation* 13.2, pp. 251–263. DOI: 10.1109/70.563647 (cited on p. 29).

Biber, P. and W. Strasser (2003). "The normal distributions transform: a new approach to laser scan matching". en. In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453).* Vol. 3. Las Vegas, Nevada, USA: IEEE, pp. 2743–2748. ISBN: 978-0-7803-7860-5. DOI: 10.1109/IROS.2003.1249285 (cited on p. 18).

Bonnifait, P. and C. Zinoune (Jan. 2021). "Introduction aux techniques de navigation autonome pour les véhicules intelligents". In: Technologies de l'information, pp. 1–20. DOI: 10.51257/a-v1-s7819 (cited on p. 174).

Caesar, H., V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom (2019). "nuScenes: A multimodal dataset for autonomous driving". In: *arXiv preprint arXiv:1903.11027* (cited on p. 127).

Canny, J. (1986). "A computational approach to edge detection". In: *IEEE Transactions on pattern analysis and machine intelligence* 6, pp. 679–698 (cited on p. 25).

Cao, B., C.-N. Ritter, D. Gohring, and R. Rojas (Sept. 2020). "Accurate Localization of Autonomous Vehicles Based on Pattern Matching and Graph-Based Optimization in Urban Environments". en. In: *IEEE International Conference on Intelligent Transportation Systems*. Rhodes, Greece. ISBN: 978-1-72814-149-7. DOI: 10.1109/ITSC45102.2020.9294299 (cited on p. 98).

Carcanague, S., O. Julien, W. Vigneau, and C. Macabiau (Sept. 2011). "Undifferenced ambiguity resolution applied to RTK". In: *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011).* Portland, United States, pp 663–678. URL: https://enac.hal.science/hal-01022490 (cited on p. 91).

Censi, A. (May 2008). "An ICP variant using a point-to-line metric". en. In: *2008 IEEE International Conference on Robotics and Automation*. Pasadena, CA, USA: IEEE, pp. 19–25. ISBN: 978-1-4244-1646-2. DOI: 10.1109/ROBOT.2008.4543181 (cited on p. 17).

Chachuła, K., J. Lyskawa, B. Olber, P. Fratczak, A. Popowicz, and K. Radlak (2023). "Combating noisy labels in object detection datasets". In: *arXiv preprint arXiv:2211.13993* (cited on pp. 61, 62).

Chadwick, S. and P. Newman (2019). "Training Object Detectors With Noisy Data". In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. DOI: 10.1109/IVS.2019.8814137 (cited on p. 61).

Chalvatzaras, A., I. Pratikakis, and A. A. Amanatiadis (2023). "A Survey on Map-Based Localization Techniques for Autonomous Vehicles". In: *IEEE Transactions on Intelligent Vehicles* 8.2, pp. 1574–1596. DOI: 10.1109/TIV.2022.3192102 (cited on p. 21).

Chen, S. and V. Fremont (n.d.). "A Loosely Coupled Vision-LiDAR Odometry Using Covariance Intersection Filtering". en. In: (), p. 6 (cited on p. 19).

Chen, Y. and G. Medioni (Apr. 1992). "Object modelling by registration of multiple range images". en. In: *Image and Vision Computing* 10.3, pp. 145–155. ISSN: 02628856. DOI: 10.1016/0262-8856(92)90066-C (cited on p. 17).

Chghaf, M., S. Rodriguez, and A. E. Ouardi (May 2022). "Camera, LiDAR and Multi-modal SLAM Systems for Autonomous Ground Vehicles: a Survey". en. In: *Journal of Intelligent & Robotic Systems* 105.1, p. 2. ISSN: 0921-0296, 1573-0409. DOI: 10.1007/s10846-022-01582-8 (cited on p. 19).

Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016). "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cited on p. 26).

Denœux, T., N. El Zoghby, V. Cherfaoui, and A. Jouglet (2014). "Optimal Object Association in the Dempster–Shafer Framework". In: *IEEE Transactions on Cybernetics* 44.12, pp. 2521–2531. DOI: 10.1109/TCYB.2014.2309632 (cited on p. 92).

Doherty, K. J., D. P. Baxter, E. Schneeweiss, and J. J. Leonard (May 2020). "Probabilistic Data Association via Mixture Models for Robust Semantic SLAM". en. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, pp. 1098–1104. ISBN: 978-1-72817-395-5. DOI: 10.1109/ICRA40945.2020.9197382 (cited on p. 92).

Dong, H., X. Chen, and C. Stachniss (Aug. 2021). "Online Range Image-based Pole Extractor for Long-term LiDAR Localization in Urban Environments". In: *2021 European Conference on Mobile Robots (ECMR)*. IEEE. DOI: 10.1109/ecmr50962.2021.9568850 (cited on p. 24).

Dong, H., X. Chen, S. Särkkä, and C. Stachniss (2023). "Online pole segmentation on range images for long-term LiDAR localization in urban environments". In: *Robotics and Autonomous Systems*. ISSN: 0921-8890. DOI: https://doi.org/10.1016/j.robot.2022.104283 (cited on pp. 34, 126, 127).

Dubé, R., D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena (May 2017). "SegMatch: Segment based loop-closure for 3D point clouds". en. In: *IEEE International Conference on Robotics and Automation*, pp. 5266–5272. DOI: 10.1109/ICRA.2017.7989618 (cited on p. 20).

Duda, R. O. and P. E. Hart (1972). "Use of the Hough transformation to detect lines and curves in pictures". In: *Commun. ACM* 15.1, 11–15. ISSN: 0001-0782. DOI: 10.1145/361237.361242 (cited on p. 25).

Ebrahimi Soorchaei, B., M. Razzaghpour, R. Valiente, A. Raftari, and Y. P. Fallah (Oct. 2022). "High-Definition Map Representation Techniques for Automated Vehicles". en. In: *Electronics* 11.20, p. 3374. ISSN: 2079-9292. DOI: 10.3390/electronics11203374 (cited on p. 14).

Elghazaly, G., R. Frank, S. Harvey, and S. Safko (2023). "High-Definition Maps: Comprehensive Survey, Challenges, and Future Perspectives". en. In: *IEEE Open Journal of Intelligent Transportation Systems* 4, pp. 527–550. ISSN: 2687-7813. DOI: 10.1109/OJITS.2023.3295502 (cited on pp. 12, 14, 20).

Ertler, C., J. Mislej, T. Ollmann, L. Porzi, G. Neuhold, and Y. Kuang (2020). *The Mapillary Traffic Sign Dataset for Detection and Classification on a Global Scale*. arXiv: 1909.04422 [cs.CV] (cited on p. 34).

Ester, M., H.-P. Kriegel, J. Sander, and X. Xu (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*. Vol. 96. 34, pp. 226–231 (cited on p. 24).

Frisch, G., P. Xu, and E. Stawiarski (2018). "High Integrity Lane Level Localization Using Multiple Lane Markings Detection and Horizontal Protection Levels". In: *15th International Conference on Control, Automation, Robotics and Vision*, pp. 1496–1501. DOI: 10.1109/ICARCV.2018.8581278 (cited on p. 99).

Geiger, A., P. Lenz, and R. Urtasun (2012). "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *2012 IEEE Conf. on Computer Vision and Pattern Recognition*. DOI: 10.1109/CVPR.2012.6248074 (cited on p. 127).

Geng, J., J. Guo, X. Meng, and K. Gao (Jan. 2020). "Speeding up PPP ambiguity resolution using triple-frequency GPS/BeiDou/Galileo/QZSS data". In: *Journal of Geodesy* 94.1, 6, p. 6. DOI: 10.1007/s00190-019-01330-110.31223/osf.io/fhdq8 (cited on p. 92).

Geng, Y., Z. Wang, L. Jia, Y. Qin, Y. Chai, K. Liu, and L. Tong (2023). "3DGraphSeg: A Unified Graph Representation- Based Point Cloud Segmentation Framework for Full-Range High-Speed Railway Environments". In: *IEEE Transactions on Industrial Informatics* 19.12, pp. 11430–11443. DOI: 10.1109/TII.2023.3246492 (cited on p. 126).

Ghallabi, F., G. El-Haj-Shhade, M.-A. Mittet, and F. Nashashibi (June 2019). "LIDAR-Based road signs detection For Vehicle Localization in an HD Map". en. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. Paris, France: IEEE, pp. 1484–1490. ISBN: 978-1-72810-560-4. DOI: 10.1109/IVS.2019.8814029 (cited on p. 24).

Ghallabi, F., F. Nashashibi, G. El-Haj-Shhade, and M.-A. Mittet (Nov. 2018). "LIDAR-Based Lane Marking Detection For Vehicle Positioning in an HD Map". en. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI: IEEE, pp. 2209–2214. ISBN: 978-1-72810-321-1 978-1-72810-323-5. DOI: 10.1109/ITSC.2018.8569951 (cited on pp. 24, 25).

Girshick, R., J. Donahue, T. Darrell, and J. Malik (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*. arXiv: 1311.2524 [cs.CV] (cited on p. 60).

Gonzalez, Á., M. Á. Garrido, D. F. Llorca, M. Gavilan, J. P. Fernandez, P. F. Alcantarilla, I. Parra, F. Herranz, L. M. Bergasa, M. Á. Sotelo, and P. Revenga De Toro

(June 2011). "Automatic Traffic Signs and Panels Inspection System Using Computer Vision". en. In: *IEEE Transactions on Intelligent Transportation Systems* 12.2, pp. 485–499. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2010.2098029 (cited on p. 25).

Harris, C. G. and M. J. Stephens (1988). "A Combined Corner and Edge Detector". In: *Alvey Vision Conference* (cited on p. 25).

Hassani, A. and M. Joerger (2023). "Analytical and Empirical Navigation Safety Evaluation of a Tightly Integrated Lidar/IMU Using Return-Light Intensity". en. In: *NAVIGATION: Journal of the Institute of Navigation* 70.4, navi.623. ISSN: 0028-1522, 2161-4296. DOI: 10.33012/navi.623 (cited on p. 28).

He, K., G. Gkioxari, P. Dollár, and R. Girshick (2017). "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322 (cited on p. 61).

Herb, M., M. Lemberger, M. M. Schmitt, A. Kurz, T. Weiherer, N. Navab, and F. Tombari (Sept. 2021). "Semantic Image Alignment for Vehicle Localization". en. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Prague, Czech Republic: IEEE, pp. 1124–1131. ISBN: 978-1-66541-714-3. DOI: 10.1109/IROS51168.2021.9636517 (cited on p. 18).

Hofstetter, I., M. Springer, F. Ries, and M. Haueis (2021). "Constellation Codebooks for Reliable Vehicle Localization". en. In: *Pattern Recognition*. Vol. 12544, pp. 303–315. ISBN: 978-3-030-71277-8 978-3-030-71278-5. DOI: 10.1007/978-3-030-71278-5_22 (cited on p. 98).

Hofstetter, I., M. Sprunk, F. Schuster, F. Ries, and M. Haueis (June 2019). "On Ambiguities in Feature-Based Vehicle Localization and their A Priori Detection in Maps". en. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. Paris, France: IEEE, pp. 1192–1198. ISBN: 978-1-72810-560-4. DOI: 10.1109/IVS.2019.8813978 (cited on p. 98).

Hornung, A., K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard (Apr. 2013). "OctoMap: an efficient probabilistic 3D mapping framework based on octrees". en. In: *Autonomous Robots* 34.3, pp. 189–206. ISSN: 0929-5593, 1573-7527. DOI: 10.1007/s10514-012-9321-0 (cited on p. 17).

Hoshino, Y., L. Yang, and S. Suzuki (2016). "Self-localization method using a single omni-directional camera based on landmark positions and arrangement". In: *IEEE/SICE International Symposium on System Integration*, pp. 580–585. DOI: 10.1109/SII.2016.7844061 (cited on p. 100).

Houben, S., J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel (2013). "Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark". In: *International Joint Conference on Neural Networks*. 1288 (cited on p. 34).

Hrustic, E. and D. Vivet (Dec. 2020). "Using Traffic Signs as Landmarks in Object-oriented EKF-SLAM". en. In: *International Conference on Control, Automation, Robotics and Vision*. Shenzhen, China, pp. 273–280. ISBN: 978-1-72817-709-0. DOI: 10.1109/ICARCV50220.2020.9305318 (cited on p. 20).

Huang, H., F. Maire, and N. Keeratiprano (June 2007). "Bearing-only Simultaneous Localization and Mapping for Vision-Based Mobile Robots". en. In: *Vision Systems: Applications*. Ed. by Goro Obinata and Ashish Dutt. I-Tech Education and Publishing. ISBN: 978-3-902613-01-1. DOI: 10.5772/4996 (cited on p. 19).

Héroux, P. and J. Kouba (Jan. 2001). "GPS precise point positioning using IGS orbit products". en. In: *Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy* 26.6-8, pp. 573–578. ISSN: 14641895. DOI: 10.1016/S1464-1895(01)00103-X (cited on p. 92).

Javanmardi, E., M. Javanmardi, Y. Gu, and S. Kamijo (2018). "Factors to Evaluate Capability of Map for Vehicle Localization". en. In: *IEEE Access* 6, pp. 49850–49867. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2868244 (cited on p. 16).

Jensfelt, P., D. Kragic, J. Folkesson, and M. Bjorkman (2006). "A framework for vision based bearing only 3D SLAM". en. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* Orlando, FL, USA: IEEE, pp. 1944–1950. ISBN: 978-0-7803-9505-3. DOI: 10.1109/ROBOT.2006.1641990 (cited on pp. 19, 100).

Jiménez, V., J. Godoy, A. Artuñedo, and J. Villagra (2021). "Ground Segmentation Algorithm for Sloped Terrain and Sparse LiDAR Point Cloud". In: *IEEE Access*. DOI: 10.1109/ACCESS.2021.3115664 (cited on pp. 39, 47, 127).

Joerger, M., G. D. Arana, M. Spenko, and B. Pervan (Nov. 2017). "Landmark Data Selection and Unmapped Obstacle Detection in Lidar-Based Navigation". en. In: Portland, Oregon, pp. 1886–1903. DOI: 10.33012/2017.15406 (cited on p. 28).

Joerger, M., M. Jamoom, M. Spenko, and B. Pervan (2016). "Integrity of laser-based feature extraction and data association". In: *2016 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 557–571. DOI: 10.1109/PLANS.2016.7479746 (cited on p. 28).

Katiyar, S. K. and P. V. Arun (2014). *Comparative analysis of common edge detection techniques in context of object extraction*. arXiv: 1405.6132 [cs.CV]. URL: https://arxiv.org/abs/1405.6132 (cited on p. 25).

Kerbl, B., G. Kopanas, T. Leimkühler, and G. Drettakis (2023). "3D Gaussian Splatting for Real-Time Radiance Field Rendering." In: *ACM Trans. Graph.* 42.4, pp. 139–1 (cited on p. 154).

Konrad, T., J.-J. Gehrt, J. Lin, R. Zweigel, and D. Abel (2018). "Advanced state estimation for navigation of automated vehicles". en. In: *Annual Reviews in Control* 46, pp. 181–195. ISSN: 13675788. DOI: 10.1016/j.arcontrol.2018.09.002 (cited on p. 98).

Kuhn, H. W (1955). "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2, pp. 83–97 (cited on p. 95).

Lategahn, H. and C. Stiller (2014). "Vision-Only Localization". In: *IEEE Transactions on Intelligent Transportation Systems* 15.3, pp. 1246–1257. DOI: 10.1109/TITS.2014.2298492 (cited on p. 19).

Lee, D.-H. (2013). "Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks". In: *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)* (cited on p. 34).

Lee, W. H., K. Jung, C. Kang, and H. S. Chang (2021). "Semi-Automatic Framework for Traffic Landmark Annotation". In: *IEEE Open Journal of Intelligent Transportation Systems*. DOI: 10.1109/OJITS.2021.3053337 (cited on p. 34).

Lehtomäki, M., A. Jaakkola, J. Hyyppä, A. Kukko, and H. Kaartinen (2010). "Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data". In: *Remote Sensing* 2.3, pp. 641–664. ISSN: 2072-4292. DOI: 10.3390/rs2030641 (cited on p. 24).

Lemaire, T., C. Berger, I.-K. Jung, and S. Lacroix (July 2007). "Vision-Based SLAM: Stereo and Monocular Approaches". en. In: *International Journal of Computer Vision* 74.3, pp. 343–364. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-007-0042-3 (cited on pp. 19, 100).

Li, F., P. Bonnifait, J. Ibanez-Guzman, and C. Zinoune (June 2017). "Lane-level map-matching with integrity on high-definition maps". en. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, pp. 1176–1181. ISBN: 978-1-5090-4804-5. DOI: 10.1109/IVS.2017.7995872 (cited on p. 22).

Li, L., M. Yang, C. Wang, and B. Wang (June 2016a). "Road DNA based localization for autonomous vehicles". en. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. Gotenburg, Sweden: IEEE, pp. 883–888. ISBN: 978-1-5090-1821-5. DOI: 10.1109/IVS.2016.7535492 (cited on p. 17).

Li, L., M. Yang, L. Weng, and C. Wang (2021). "Robust Localization for Intelligent Vehicles Based on Pole-Like Features Using the Point Cloud". en. In: *IEEE Transactions on Automation Science and Engineering*, pp. 1–14. ISSN: 1545-5955, 1558-3783. DOI: 10.1109/TASE.2020.3048333 (cited on pp. 24, 97).

Li, X., J. Huang, X. Li, Z. Shen, J. Han, L. Li, and B. Wang (Dec. 2022). "Review of PPP–RTK: achievements, challenges, and opportunities". en. In: *Satellite Navigation* 3.1, p. 28. ISSN: 2662-1363. DOI: 10.1186/s43020-022-00089-9 (cited on p. 92).

Li, Y., L. Chen, H. Huang, X. Li, W. Xu, L. Zheng, and J. Huang (2016b). "Nighttime lane markings recognition based on Canny detection and Hough transform". In: *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 411–415. DOI: 10.1109/RCAR.2016.7784064 (cited on p. 25).

Liao, Y., J. Xie, and A. Geiger (2022). "KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D". In: *Pattern Analysis and Machine Intelligence (PAMI)* (cited on p. 127).

Lin, J., C. Zheng, W. Xu, and F. Zhang (Oct. 2021). "R $ˆ2$ LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping". en. In: *IEEE Robotics and Automation Letters* 6.4, pp. 7469–7476. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3095515 (cited on p. 19).

Lin, T.-Y., M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár (2015). *Microsoft COCO: Common Objects in Context*. arXiv: 1405.0312 [cs.CV] (cited on p. 34).

Lopez, L. D. and O. Fuentes (2007). "Color-Based Road Sign Detection and Tracking". en. In: *Image Analysis and Recognition*. Vol. 4633. ISSN: 0302-9743,

1611-3349 Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1138–1147. ISBN: 978-3-540-74258-6 978-3-540-74260-9. DOI: 10.1007/978-3-540-74260-9_101 (cited on p. 25).

Lu, W., E. Seignez, F. S. A. Rodriguez, and R. Reynaud (Dec. 2014). "Lane marking based vehicle localization using particle filter and multi-kernel estimation". en. In: *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. Singapore: IEEE, pp. 601–606. ISBN: 978-1-4799-5199-4. DOI: 10.1109/ICARCV.2014.7064372 (cited on p. 25).

Ma, X., C. Qin, H. You, H. Ran, and Y. Fu (2022). *Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework*. arXiv: 2202.07123 [cs.CV] (cited on p. 126).

Magnusson, M., A. Lilienthal, and T. Duckett (Oct. 2007). "Scan registration for autonomous mining vehicles using 3D-NDT". en. In: *Journal of Field Robotics* 24.10, pp. 803–827. ISSN: 1556-4959, 1556-4967. DOI: 10.1002/rob.20204 (cited on p. 18).

Magnusson, M., A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg (2009). "Evaluation of 3D registration reliability and speed - A comparison of ICP and NDT". In: *2009 IEEE International Conference on Robotics and Automation*, pp. 3907–3912. DOI: 10.1109/ROBOT.2009.5152538 (cited on p. 18).

Mammeri, A., A. Boukerche, and G. Lu (2014). "Lane detection and tracking system based on the MSER algorithm, hough transform and kalman filter". In: *Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. MSWiM '14. Montreal, QC, Canada: Association for Computing Machinery, 259–266. ISBN: 9781450330305. DOI: 10.1145/2641798.2641807 (cited on p. 25).

Maturana, D. and S. Scherer (2015). "VoxNet: A 3D Convolutional Neural Network for real-time object recognition". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928. DOI: 10.1109/IROS.2015.7353481 (cited on p. 126).

Mendes, E., P. Koch, and S. Lacroix (Oct. 2016). "ICP-based pose-graph SLAM". en. In: *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. Lausanne, Switzerland: IEEE, pp. 195–200. ISBN: 978-1-5090-4349-1. DOI: 10.1109/SSRR.2016.7784298 (cited on p. 19).

Mildenhall, B., P. P. Srinivasan, M. Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng (2021). "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *Communications of the ACM* 65.1, pp. 99–106 (cited on p. 154).

Milioto, A., I. Vizzo, J. Behley, and C. Stachniss (2019). "RangeNet ++: Fast and Accurate LiDAR Semantic Segmentation". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4213–4220. DOI: 10.1109/IROS40897.2019.8967762 (cited on p. 126).

Missaoui, B., M. Noizet, and P. Xu (2023). "Map-aided annotation for pole base detection". In: *IEEE Intelligent Vehicles Symposium (IV)*. DOI: 10.1109/IV55152.2023.10186774. URL: https://arxiv.org/abs/2403.01868 (cited on pp. 5, 62).

Mur-Artal, R., J. M. M. Montiel, and J. D. Tardos (Oct. 2015). "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". en. In: *IEEE Transactions on Robotics* 31.5, pp. 1147–1163. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2015.2463671 (cited on p. 19).

Mur-Artal, R. and J. D. Tardos (Oct. 2017). "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". en. In: *IEEE Transactions on Robotics* 33.5. arXiv:1610.06475 [cs], pp. 1255–1262. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2017.2705103 (cited on p. 19).

Nagai, K., M. Spenko, R. Henderson, and B. Pervan (2024). "Fault-Free Integrity of Urban Driverless Vehicle Navigation with Multi-Sensor Integration: A Case Study in Downtown Chicago". en. In: *NAVIGATION: Journal of the Institute of Navigation* 71.1, navi.631. ISSN: 0028-1522, 2161-4296. DOI: 10.33012/navi.631 (cited on p. 28).

Noizet, M., P. Xu, and P. Bonnifait (2023). "Pole-Based Vehicle Localization with Vector Maps: A Camera-LiDAR Comparative Study". In: *International Conf. on Intelligent Transportation Systems (ITSC)*. DOI: 10.1109/ITSC57777.2023.10422577 (cited on pp. 5, 134).

— (2024). "Automatic Image Annotation for Mapped Features Detection". In: *International Conf. on Intelligent Robots and Systems (IROS)*, pp. 9367–9373. DOI: 10.1109/IROS58592.2024.10801773 (cited on pp. 5, 48, 53, 55, 69).

Northcutt, C., L. Jiang, and I. Chuang (2021). "Confident learning: Estimating uncertainty in dataset labels". In: *Journal of Artificial Intelligence Research* (cited on p. 62).

Oh, S.-I. and H.-B. Kang (2016). "Fast Occupancy Grid Filtering Using Grid Cell Clusters From LIDAR and Stereo Vision Sensor Data". In: *IEEE Sensors Journal* 16.19, pp. 7258–7266. DOI: 10.1109/JSEN.2016.2598600 (cited on p. 17).

Pauls, J.-H., K. Petek, F. Poggenhans, and C. Stiller (Oct. 2020). "Monocular Localization in HD Maps by Combining Semantic Segmentation and Distance Transform". en. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, pp. 4595–4601. ISBN: 978-1-72816-212-6. DOI: 10.1109/IROS45743.2020.9341003 (cited on p. 92).

Pettersson, N., L. Petersson, and L. Andersson (June 2008). "The histogram feature - a resource-efficient Weak Classifier". en. In: *2008 IEEE Intelligent Vehicles Symposium*. Eindhoven, Netherlands: IEEE, pp. 678–683. ISBN: 978-1-4244-2568-6. DOI: 10.1109/IVS.2008.4621174 (cited on p. 25).

Poggenhans, F., J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr (2018). "Lanelet2: A high-definition map framework for the future of automated driving". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1672–1679. DOI: 10.1109/ITSC.2018.8569929 (cited on p. 20).

Qi, C. R., L. Yi, H. Su, and L. J. Guibas (2017). "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". In: *arXiv preprint arXiv:1706.02413* (cited on pp. 126, 127).

Qi, C. R., Y. Zhou, M. Najibi, P. Sun, K. Vo, B. Deng, and D. Anguelov (2021). "Offboard 3D Object Detection from Point Cloud Sequences". In: *2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. DOI: 10.1109/CVPR46437.2021.00607 (cited on p. 34).

Radosavovic, I., P. Dollár, R. Girshick, G. Gkioxari, and K. He (2018). "Data Distillation: Towards Omni-Supervised Learning". In: *2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. DOI: 10.1109/CVPR.2018.00433 (cited on p. 34).

Reed, S., H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich (2014). "Training deep neural networks on noisy labels with bootstrapping". In: *arXiv preprint arXiv:1412.6596* (cited on p. 62).

Revilloud, M., D. Gruyer, and E. Pollard (June 2013). "An improved approach for robust road marking detection and tracking applied to multi-lane estimation". en. In: *2013 IEEE Intelligent Vehicles Symposium (IV)*. Gold Coast City, Australia: IEEE, pp. 783–790. ISBN: 978-1-4673-2755-8 978-1-4673-2754-1. DOI: 10.1109/IVS.2013.6629562 (cited on p. 25).

Riveiro, B., L. Diaz-Vilarino, B. Conde-Carnero, M. Soilan, and P. Arias (Jan. 2016). "Automatic Segmentation and Shape-Based Classification of Retro-Reflective Traffic Signs from Mobile LiDAR Data". en. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.1, pp. 295–303. ISSN: 1939-1404, 2151-1535. DOI: 10.1109/JSTARS.2015.2461680 (cited on p. 24).

Rodríguez-Cuenca, B., S. García-Cortés, C. Ordóñez, and M. Alonso (Sept. 2015). "Automatic Detection and Classification of Pole-Like Objects in Urban Point Cloud Data Using an Anomaly Detection Algorithm". en. In: *Remote Sensing* 7.10, pp. 12680–12703. ISSN: 2072-4292. DOI: 10.3390/rs71012680 (cited on p. 24).

Saarinen, J., H. Andreasson, T. Stoyanov, J. Ala-Luhtala, and A. J. Lilienthal (2013). "Normal Distributions Transform Occupancy Maps: Application to large-scale online 3D mapping". In: *2013 IEEE International Conference on Robotics and Automation*, pp. 2233–2238. DOI: 10.1109/ICRA.2013.6630878 (cited on p. 18).

Sathish, P. and D. Bharathi (2016). "Automatic Road Sign Detection and Recognition Based on SIFT Feature Matching Algorithm". en. In: *Proceedings of the International Conference on Soft Computing Systems*. Vol. 398. Series Title: Advances in Intelligent Systems and Computing. New Delhi: Springer India, pp. 421–431. ISBN: 978-81-322-2672-7 978-81-322-2674-1. DOI: 10.1007/978-81-322-2674-1_39 (cited on p. 25).

Schaefer, A., D. Buscher, J. Vertens, L. Luft, and W. Burgard (Sept. 2019). "Long-Term Urban Vehicle Localization Using Pole Landmarks Extracted from 3-D Lidar Scans". en. In: *2019 European Conference on Mobile Robots (ECMR)*. Prague, Czech Republic: IEEE, pp. 1–7. ISBN: 978-1-72813-605-9. DOI: 10.1109/ECMR.2019.8870928 (cited on p. 24).

Schlichting, A. and U. Feuerhake (June 2018). "Global Vehicle Localization by Sequence Analysis Using LiDAR Features Derived by an Autoencoder". en. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. Changshu: IEEE, pp. 656–661. ISBN: 978-1-5386-4452-2. DOI: 10.1109/IVS.2018.8500358 (cited on p. 18).

Schubert, M., T. Riedlinger, K. Kahl, D. Kröll, S. Schoenen, S. Šegvić, and M. Rottmann (Mar. 2023). *Identifying Label Errors in Object Detection Datasets by Loss Inspection*. en. arXiv:2303.06999 [cs] (cited on p. 78).

Sefati, M., M. Daum, B. Sondermann, K. D. Kreiskother, and A. Kampker (June 2017). "Improving vehicle localization using semantic and pole-like landmarks". en. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. Los Angeles, CA, USA: IEEE, pp. 13–19. ISBN: 978-1-5090-4804-5. DOI: 10.1109/IVS.2017.7995692 (cited on p. 24).

Segal, A. V., D. Hähnel, and S. Thrun (2009). "Generalized-ICP". In: *Robotics: Science and Systems*. DOI: 10.15607/RSS.2009.V.021 (cited on p. 17).

Shao, W., S. Vijayarangan, C. Li, and G. Kantor (Feb. 2019). *Stereo Visual Inertial LiDAR Simultaneous Localization and Mapping*. en. arXiv:1902.10741 [cs] (cited on p. 19).

Smith, R., M. Self, and P. Cheeseman (1988). "Estimating Uncertain Spatial Relationships in Robotics". In: *Uncertainty in Artificial Intelligence*. Ed. by J. F. Lemmer and L. N. Kanal. Vol. 5. Machine Intelligence and Pattern Recognition. North-Holland, pp. 435–461. DOI: https://doi.org/10.1016/B978-0-444-70396-5.50042-X (cited on p. 19).

Sohn, K., Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister (2020). *A Simple Semi-Supervised Learning Framework for Object Detection* (cited on p. 34).

Steinke, N., C.-N. Ritter, D. Goehring, and R. Rojas (Apr. 2021). "Robust LiDAR Feature Localization for Autonomous Vehicles Using Geometric Fingerprinting on Open Datasets". en. In: *IEEE Robotics and Automation Letters* 6.2, pp. 2761–2767. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3062354 (cited on p. 98).

Stumberg, L. von, P. Wenzel, N. Yang, and D. Cremers (Oct. 2020). *LM-Reloc: Levenberg-Marquardt Based Direct Visual Relocalization*. en. arXiv:2010.06323 [cs] (cited on p. 18).

Subirana, J.S., M. Hernández-Pajares, J.M.J. Zornoza, European Space Agency, and K. Fletcher (2013). *GNSS Data Processing*. ESA TM vol. 1. ESA Communications. ISBN: 9789292218867 (cited on pp. 89, 90, 92, 173).

Sun, C., J. M. U. Vianney, Y. Li, L. Chen, L. Li, F.-Y. Wang, A. Khajepour, and D. Cao (2020). "Proximity based automatic data annotation for autonomous driving". In: *IEEE/CAA Journal of Automatica Sinica*. DOI: 10.1109/JAS.2020.1003033 (cited on p. 34).

Tam, G. K. L., Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin (July 2013). "Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid". en. In: *IEEE Transactions on Visualization and Computer Graphics* 19.7, pp. 1199–1217. ISSN: 1077-2626. DOI: 10.1109/TVCG.2012.310 (cited on p. 17).

Tessier, C., C. Debain, R. Chapuis, and F. Chausse (July 2010). "Map Aided Localization and vehicle guidance using an active landmark search". en. In: *Information Fusion* 11.3, pp. 283–296. ISSN: 15662535. DOI: 10.1016/j.inffus.2009.09.006 (cited on p. 23).

Timofte, R., K. Zimmermann, and L. Van Gool (Apr. 2014). "Multi-view traffic sign detection, recognition, and 3D localisation". en. In: *Machine Vision and Applications* 25.3, pp. 633–647. ISSN: 0932-8092, 1432-1769. DOI: 10.1007/s00138-011-0391-3 (cited on pp. 34, 61).

Tsai, D., J. S. Berrio, M. Shan, E. Nebot, and S. Worrall (2023). "MS3D++: Ensemble of Experts for Multi-Source Unsupervised Domain Adaptation in 3D Object Detection". In: *arXiv preprint arXiv:2308.05988* (cited on pp. 35, 127).

Wali, S., M. Abdullah, M. A. Hannan, A. Hussain, S. Samad, P. J. Ker, and M. Mansor (May 2019). "Vision-Based Traffic Sign Detection and Recognition Systems: Current Trends and Challenges". In: *Sensors* 19, p. 2093. DOI: 10.3390/s19092093 (cited on p. 25).

Wang, C.-Y., A. Bochkovskiy, and H.-Y. M. Liao (2022). "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". In: *arXiv preprint arXiv::2207.02696* (cited on pp. 60, 61).

Wang, J., K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao (2021). "Deep High-Resolution Representation Learning for Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2020.2983686 (cited on p. 44).

Wang, L., Y. Zhang, and J. Wang (July 2017). "Map-Based Localization Method for Autonomous Vehicles Using 3D-LIDAR * *This work is supported in part by the National Natural Science Foundation of China under Grant No. 61473209." en. In: *IFAC-PapersOnLine* 50.1, pp. 276–281. ISSN: 24058963. DOI: 10.1016/j.ifacol.2017.08.046 (cited on p. 24).

Wassaf, H., K. Bernazzani, P. Gandhi, J. Lu, K. Van Dyke, K. Shallberg, S. Ericson, J. Flake, and M. Herman (Oct. 2021). "Highly Automated Vehicle Absolute Positioning Using LiDAR Unique Signatures". en. In: St. Louis, Missouri, pp. 22–52. DOI: 10.33012/2021.17878 (cited on p. 27).

Wei, P., X. Wang, and Y. Guo (Aug. 2020). "3D-LIDAR Feature Based Localization for Autonomous Vehicles". en. In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. Hong Kong, Hong Kong: IEEE, pp. 288–293. ISBN: 978-1-72816-904-0. DOI: 10.1109/CASE48305.2020.9216959 (cited on p. 24).

Welte, A., P. Xu, and P. Bonnifait (2019). "Four-Wheeled Dead-Reckoning Model Calibration using RTS Smoothing". In: *2019 International Conference on Robotics and Automation (ICRA)*, pp. 312–318. DOI: 10.1109/ICRA.2019.8794270 (cited on p. 106).

Welte, A., P. Xu, P. Bonnifait, and C. Zinoune (2020). "Improved Data Association Using Buffered Pose Adjustment for Map-Aided Localization". In: *IEEE Robotics and Automation Letters* 5.4, pp. 6334–6341. DOI: 10.1109/LRA.2020.3013856 (cited on p. 92).

Williams, B., M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardos (2008). "An image-to-map loop closing method for monocular SLAM". In: *2008 IEEE/RSJ*

International Conference on Intelligent Robots and Systems, pp. 2053–2059. DOI: 10.1109/IROS.2008.4650996 (cited on p. 20).

Wu, J. T., S. C. Wu, G. A. Hajj, W. I. Bertiger, and S. M. Lichten (Aug. 1992). "Effects of antenna orientation on GPS carrier phase". In: *Astrodynamics 1991*. Ed. by Peter Blumer, pp. 1647–1660 (cited on p. 91).

Xu, X., J. Jin, S. Zhang, L. Zhang, S. Pu, and Z. Chen (May 2019). "Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry". en. In: *Future Generation Computer Systems* 94, pp. 381–391. ISSN: 0167739X. DOI: 10.1016/j.future.2018.11.027 (cited on p. 25).

Yang, L., B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao (2024). "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data". In: *CVPR* (cited on p. 154).

Yang, Q., X. Wei, B. Wang, X.-S. Hua, and L. Zhang (2021). "Interactive Self-Training with Mean Teachers for Semi-supervised Object Detection". In: *2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. DOI: 10.1109/CVPR46437.2021.00588 (cited on p. 34).

Yu, F., H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell (2020). "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning". In: *IEEE/CVF International Conf. on Computer Vision and Pattern Recognition (ICCV)* (cited on pp. 26, 27, 44).

Yu, F., A Seff, Y Zhang, S. Song, T. Funkhouser, and J. Xiao (2016). *LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop* (cited on p. 34).

Zaklouta, F. and B. Stanciulescu (Dec. 2012). "Real-Time Traffic-Sign Recognition Using Tree Classifiers". en. In: *IEEE Transactions on Intelligent Transportation Systems* 13.4, pp. 1507–1514. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2012.2225618 (cited on p. 25).

Zermas, D., I. Izzat, and N. Papanikolopoulos (2017). "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications". In: *IEEE International Conference on Robotics and Automation*, pp. 5067–5073. DOI: 10.1109/ICRA.2017.7989591 (cited on p. 128).

Zhang, J. and S. Singh (July 2014). "LOAM: Lidar Odometry and Mapping in Real-time". en. In: *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation. ISBN: 978-0-9923747-0-9. DOI: 10.15607/RSS.2014.X.007 (cited on p. 19).

Zhang, R., Y. Wu, W. Jin, and X. Meng (Aug. 2023). "Deep-Learning-Based Point Cloud Semantic Segmentation: A Survey". en. In: *Electronics* 12.17, p. 3642. ISSN: 2079-9292. DOI: 10.3390/electronics12173642 (cited on p. 126).

Zhou, J., Y. Xiong, C. Chiu, F. Liu, and X. Gong (2023). *SAT: Size-Aware Transformer for 3D Point Cloud Semantic Segmentation*. arXiv: 2301.06869 [cs.CV] (cited on p. 126).

Zhou, Y. and O. Tuzel (2017). *VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection*. arXiv: 1711.06396 [cs.CV] (cited on p. 126).

Zhu, X., H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin (2020). "Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation". In: *arXiv preprint arXiv:2011.10033* (cited on pp. 46, 126, 129).

Ziegler, J., P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knöppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb (2014). "Making Bertha Drive—An Autonomous Journey on a Historic Route". In: *IEEE Intelligent Transportation Systems Magazine* 6.2, pp. 8–20. DOI: 10.1109/MITS.2014.2306552 (cited on p. 14).