Mondragon
Unibertsitatea

DOCTORAL THESIS

COMPUTER VISION TECHNIQUES FOR AUTONOMOUS VEHICLES APPLIED TO URBAN UNDERGROUND RAILWAY

MIKEL ETXEBERRIA GARCIA | Arrasate-Mondragón, 2022

MIKEL ETXEBERRIA GARCIA | Computer vision techniques for autonomous vehicles applied to urban underground railway

Dissertation

COMPUTER VISION TECHNIQUES FOR AUTONOMOUS VEHICLES APPLIED TO URBAN UNDERGROUND RAILWAY

Mikel Etxeberria Garcia

Supervisors Dr. Nestor Arana-Arexolaleiba
Dr. Maider Zamalloa Aquizu

A thesis submitted for the degree of

Doctoral Program in Applied Engineering

Mondragon Unibertsitatea
Electronic and Computer Sciences

May 5, 2022

ikerlan
MEMBER OF BASQUE RESEARCH
& TECHNOLOGY ALLIANCE

Mondragon
Unibertsitatea

CAF
Signalling

This work is supported in part by the Basque Government through the BIKAINTEK2018 program and CAF Signalling. In collaboration between Ikerlan, CAF Signalling, and Mondragon Unibertsitatea.

EUSKO JAURLARITZA
EKONOMIAREN GARAPEN,
JASANGARRITASUN
ETA INGURUMEN SAILA

GOBIERNO VASCO
DEPARTAMENTO DE DESARROLLO
ECONÓMICO, SOSTENIBILIDAD
Y MEDIO AMBIENTE

Familiari, betikoei eta bidean etorri direnei.

Acknowledgments

I wish to show my greatest appreciation to my supervisors, Dr. Maider Zamalloa and Dr. Nestor Arana, for all the support, implication, and guidance throughout this PhD work. It has been a journey that would not have been possible without you.

To my colleagues in Ikerlan, it has been an amazing experience to share this time together, the work time, the socialization, and the coffees. I consider you friends. I would like to thank Lorea Belategi for the kind cheers and the time taken with the corrections; Unai Diaz de Cerio for the technical help with the content related to the railway; Alejandro Solis for the assistance and making the days more enjoyable; Joanes Plazaola and Jon Zubieta for their contributions.

From CAF Signalling, thanks to Mikel Labayen for all the personal and technical support and the rest of the team for all the help and company during the long nights recording on the train.

I am deeply grateful to my family, who has been there to support me every day and encourage me throughout my studies. My parents, Jon and Mari. My brother and sister-in-law, Josu and Irati. My partner, who has the patience of a thousand, Mikel. You are the best one could have.

Also, thanks to my friends that have supported me to keep going and are always ready to take out a smile from me. Especially to Alex Abuin, you got this!

Eskerrik asko denoi, milesker benetan!

Statement of originality

I hereby declare that the research recorded in this thesis and the thesis itself were developed entirely by myself at the Signal Theory and Communications Area, Department of Electronics and Computer Science, at the University of Mondragon.

Arrasate/Mondragón, May 5, 2022

Mikel Etxeberria Garcia

Abstract

Autonomous vehicles' presence is becoming a reality in everyday life, with autonomous driving cars on the road, GOA3-GOA4 trains in the railway domain, or automated guided vehicles in the industrial domain. These autonomous systems must execute complex tasks to perceive the environment and make decisions with limited human interaction or even without human interaction. In that way, localization and motion estimation are critical tasks for the operations an autonomous vehicle must accomplish. Position information is essential to identify the vehicle context and surroundings and move or act accordingly. Computer Vision-based approaches have shown promising results in mobile robotics, drones, or autonomous cars. However, the application and evaluation of CV-based solutions are more limited in the railway domain, especially in challenging environments. In this research, a state of the art of Visual Odometry (VO) and Visual SLAM (vSLAM) algorithms is carried out. In the SOTA, the analyzed VO/vSLAM algorithms are usually evaluated in outdoor street scenarios and do not consider the challenging perception conditions that can be found in urban underground railway scenarios, with low lighting conditions and texture-less areas in tunnels and significant lighting changes between tunnels and railway platforms.

Moreover, there is no reference dataset in the VO/vSLAM community with such characteristics, raising the need to generate a proprietary dataset. Considering the lack of GPS signals in underground scenarios, a method is proposed to generate a ground truth of images and poses in underground railway scenarios. The generation process is based on synchronizing geodetic coordinates, train ATP data recorded from the radar and encoder sensors, and a railway gradient map provided by the railway infrastructure manager. Two state-of-the-art and recently proposed VO/vSLAM approaches (ORB-SLAM2 and DF-VO) have been tested in the generated proprietary datasets. These algorithms have achieved good performance in standard benchmarks such as KITTI and represent two distinct VO/vSLAM algorithm types: geometric and learning-based. However, the results show that the scenario lighting characteristics significantly affect the VO/vSLAM algorithms' performance.

In order to afford the challenging lighting conditions of the underground railway domain, the application of a data enhancement technique has been considered (EnlightenGAN). As calibration is critical for geometric VO/vSLAM algorithms, the

impact of EnlightenGAN on the camera calibration parameters is also analyzed. The results demonstrate that EnlightenGAN does not considerably affect those parameters. Besides, it improves the performance of both VO/vSLAM approaches in challenging scenarios.

Resumen

La presencia de vehículos autónomos se está convirtiendo en una realidad en la vida cotidiana, con coches de conducción autónoma en la carretera, trenes GOA3-GOA4 en el ámbito ferroviario o vehículos guiados automatizados en el ámbito industrial. Estos sistemas autónomos deben ejecutar tareas complejas para percibir el entorno y tomar decisiones con una interacción humana limitada o incluso sin ella. Siendo esto así, la localización y la estimación del movimiento son tareas críticas para las operaciones que debe realizar un vehículo autónomo. La información sobre la posición es esencial para identificar el contexto del vehículo y su entorno y moverse o actuar en consecuencia. Los enfoques basados en la visión artificial (CV) han mostrado resultados prometedores en la robótica móvil, los drones o los coches autónomos. Sin embargo, la aplicación y evaluación de las soluciones basadas en CV son más limitadas en el ámbito ferroviario, especialmente en entornos desafiantes en cuanto a características visuales. En esta investigación, se realiza un estado del arte (SOTA) de los algoritmos de Odometría Visual (VO) y SLAM Visual (vSLAM). En el SOTA, los algoritmos VO/vSLAM analizados suelen evaluarse en escenarios exteriores y no consideran las retadoras características perceptuales que pueden encontrarse en los escenarios ferroviarios subterráneos urbanos, con condiciones de baja iluminación y zonas sin texturas en los túneles y cambios de iluminación significativos entre los túneles y las estaciones ferroviarias.

Además, no existe ningún dataset de referencia en la comunidad VO/vSLAM con estas características, lo que ha planteado la necesidad de generar un conjunto de datos propio. Teniendo en cuenta la falta de señales GPS en escenarios subterráneos, se propone un método para generar un dataset de imágenes con datos verificados sobre el terreno de posiciones en escenarios ferroviarios subterráneos urbanos. El proceso de generación se basa en la sincronización de coordenadas geodésicas, los datos de ATP del tren registrados desde los sensores de radar y codificadores, y un mapa de gradiente ferroviario proporcionado por el administrador de la infraestructura ferroviaria. Se han probado dos algoritmos VO/vSLAM de última generación y recientemente propuestos (ORB-SLAM2 y DF-VO) en los dataset generados. Estos algoritmos han logrado un buen rendimiento en datasets estándar como KITTI y representan dos tipos de algoritmos VO/vSLAM distintos: geométricos y basados en el aprendizaje automático. Sin embargo, los resultados muestran que las caracterís-

ticas de iluminación del escenario afectan significativamente al rendimiento de los algoritmos VO/vSLAM.

Para afrontar las difíciles condiciones de iluminación del ámbito ferroviario subterráneo, se ha considerado la aplicación de una técnica de mejora de datos (EnlightenGAN). Como la calibración es fundamental para los algoritmos geométricos VO/vSLAM, también se ha analizado el impacto de EnlightenGAN en los parámetros de calibración de la cámara. Los resultados demuestran que EnlightenGAN no afecta considerablemente a esos parámetros. Además, mejora el rendimiento de ambos enfoques VO/vSLAM en escenarios difíciles.

Laburpena

Ibilgailu autonomoen presentzia errealitate bihurtzen ari da egunerokoan, errepidean gidatze autonomoko autoak, trenbideko GOA3-GOA4 trenak edo industria-eremuko ibilgailu automatizatuak direla eta. Sistema autonomo horiek, ataza konplexuak burutu behar dituzte ingurunea hautemateko eta erabakiak hartzeko (giza elkarreragin mugatuarekin edo interakziorik gabe). Hori horrela izanik, lokalizazioa eta mugimenduaren estimazioa eginkizun kritikoak dira ibilgailu autonomo batek egin beharreko eragiketetarako. Posizioari buruzko informazioa funtsezkoa da, ibilgailuaren testuingurua eta ingurunea identifikatzeko, eta horren arabera mugitzeko edo jarduteko. Ikusmen Artifizialean (IA) oinarritutako ikuspuntuek emaitza oparoak erakutsi dituzte robotika mugikorrean, droneetan edo auto autonomoetan. Hala ere, IAean oinarritutako irtenbideen aplikazioa eta ebaluazioa mugatuagoak dira trenbide-eremuan, batez ere, erronka bisual bat aurkezten duten inguruneetan. Ikerketa honetan, Visual Odometry (VO) eta Visual SLAM (vSLAM) algoritmoen uneko egoera (SOTA) egiten da. SOTAn, aztertutako VO/vSLAM algoritmoak, kanpoko agertokietan ebaluatu ohi dira, eta ez dituzte kontuan hartzen hiriko lurpeko trenbide-agertokietan aurki daitezkeen erronka bereizgarriak. Hala nola, tuneletan aurkitzen diren argiztapen baxuko baldintzak, testurarik gabeko eremuak eta tunel-geltokien arteko argiztapen aldaketa nabarmenaktuneletan argiztapen baxuko baldintzekin eta testurarik gabeko eremuekin eta tunelen eta trenbide-geltokien arteko argiztapen-aldaketa nabarmenekin.

Gainera, VO/vSLAM komunitatean ez dago ezaugarri horiek dituen erreferentziazko datu baserik, eta horrek berezko datu base bat egiteko beharra sortu du. Lurpeko agertokietan GPS seinalerik ez dagoela kontuan hartuta, irudi eta posizio datu base bat sortzeko metodo bat proposatzen da, lurpeko hiri-trenbide-ingurunean egiaztatutako datuekin. Sortze-prozesua, koordenatu geodesikoen sinkronizazioan, radar-sentsoreetatik eta kodifikatzaileetatik erregistratutako trenaren ATP datuetan, eta tren-azpiegituraren administratzaileak hornitutako trenbide-gradientearen mapan oinarritzen da. Punta-puntako bi VO/vSLAM algoritmo probatu dira (ORB-SLAM2 eta DF-VO) sortutako datu baseetan. Algoritmo horiek, errendimendu ona lortu dute KITTI bezalako dataset estandarretan, eta bi algoritmo mota ordezkatzen dituzte: geometrikoak eta ikaskuntzan oinarritutakoak. Hala ere, emaitzek erakutsi

dute ingurunearen argiztapen-ezaugarriek nabarmen eragiten diotela VO/vSLAM algoritmoen errendimenduari.

Lurpeko trenbidearen argiztapen-baldintza zailei aurre egiteko, datuak hobetzeko teknika bat (EnlightenGAN) aplikatzea erabaki da. Kalibrazioa VO/vSLAM algoritmo geometrikoetarako funtsezkoa denez, EnlightenGAN-ek kameraren kalibrazio-parametroetan duen eragina ere aztertu da. Emaitzek erakusten dute EnlightenGAN-ek ez diela nabarmen eragiten parametro horiei. Gainera, bi VO/vSLAM algoritmoen errendimendua hobetzen du argiztapen egoera zailetan.

Contents

List of Figures

List of Tables

Acronyms

2D Two-dimensional.

3D Three-dimensional.

ADAS Advanced Driver-assistance systems.

AI Artificial Intelligence.

API Application Programming Interface.

ATE Absolute Trajectory Error.

ATO Automatic Train Operation.

ATP Automatic Train Protection.

BNN Bayesian neural network.

CBTC Communication-based Train Control.

CNN Convolutional neural network.

CV Computer Vision.

DARPA Defense Advanced Research Projects Agency.

DoF Degrees of freedom.

DVS Dynamic Vision Sensors.

ERTMS European Rail Traffic Management System.

ETCS European Train Control System.

GAN Generative adversarial network.

GNSS Global Navigation Satellite System.

GOA Grade of automation.

HD High definition.

HDR High dynamic range.

IEEE Institute of Electrical and Electronics Engineers.

IGN Instituto Geografico Nacional.

IMU Inertial Measurement Unit.

INS Inertial Navigation System.

ITS Intelligent transportation systems.

L3 Line 3 of Euskotren in Bilbao.

Lat Latitutde.

LiDAR Light detection and ranging.

Lon Longitude.

LSTM Long short-term memory.

MoCap Motion capture system.

MRE Mean Reprojection Error.

NN Neural network.

PNG Portable network graphics.

RCNN Recurrent convolutional neural network.

RGB Red-gree-blue color model.

RMSE Root Mean Square Error.

RNN Recurrent neural network.

RPE Relative Pose Error.

SCNE Sistema Cartográfico Nacional.

SfM Structure from motion.

SLAM Simultaneous location and mapping.

SOTA State of the art.

SSIM Structural Similarity Index Measure.

UAM Uniformly Accelerated Motion.

UAV Unmanned aerial vehicle.

UTC Coordinated Universal Time.

VO Visual Odometry.

Introduction 1

One of the most promising research fields in the last years is the Computer Vision (CV). CV is the field of science and engineering that covers the camera-based system development to acquire, process, analyze and understand real-world scenes [1]. Following this, CV has several open problems that have been targeted by the research community in the last years. Some of these problems include the object detection, the image reconstruction, scene instance segmentation and localization.

Precisely, localization is one of the primary input data of many functions in autonomous robotics. A robot must understand its surrounding environment to keep knowledge of its position throughout time and achieve autonomous operations, especially navigation. Robot localization is a well-known task that continues generating new approaches and research works in the research community.

From researched robot localization techniques, the ones that include CV give the robot the capacity to localize itself using cameras as sensors. The motion of a moving camera from a sequence of images is also defined as ego-motion [2]. The autonomous localization research community started from the robotics domain to, later, focus on the localization in other sub-domains. In this context, different types of vehicles from distinct sub-domains and diverse characteristics have been studied, such as, cars [3], [4], trains [5], or lately UAVs [6].

The railway domain is also moving towards the *Intelligent Transportation Systems* (ITS) and the *Advanced Driving Assistance Systems* (ADAS) industry. This thesis targets this domain, focusing on autonomous trains driving through urban underground scenarios. A train that implements autonomous operations requires accurate localization estimation to carry out operations as precise stop operation or coupling successfully. Precise localization systems can reach a higher grade of automation [7]. Therefore, it becomes essential to implement precise and dependable train localization subsystems.

Communication-Based Train Control (CBTC) is a standard defined by the IEEE (IEEE 1474 [8]), which defines a set of performance and functional requirements for track and onboard equipment in order to enhance performance, availability, operations, and the protection of the involved systems. A CBTC system could be defined as an automatic train control system where the track and onboard subsystems are continuously communicated. Current CBTC systems, according to the standard IEC 62290-1, can be divided into pre-established Grades of Autonomy (GOA). The GOA of a train implementing any autonomous operation will have a value between 2 and 4: GOA2 for a semi-automated Train Operation, GOA3 for a driverless Train Operation, and GOA4 for an unattended Train Operation. The main two functionalities covered by those subsystems are the Automatic Train Protection (ATP) and Automatic Train Operation (ATO). ATP subsystems monitor the train speed and position to guarantee a safe train operation. On the other hand, ATO subsystems are dedicated to the operations devoted to reaching a more autonomous and efficient train driving experience, such as driving assistance tasks or automatic control of train brake and traction commands that aim to ensure that train speed is lower than the limit established by the ATP system [9].

The ERTMS/ETCS ATP train speed estimation process is based on a redundant wheel encoder and radar sensors. Using these sensors, the ATP subsystem embedded in the train estimates the train position on the track, i.e. the distance traveled from a station or a beacon of the track. Track beacon position or inter-beacon distance is predefined and known by railway infrastructure managers, even by the ATP subsystem, and therefore, the ATP train position is re-adjusted when a beacon signal is received, obtaining a precise estimation. However, the beacon system's cost must be considered, as the infrastructure cost is very high, and the deployment is slowed down [10]. However, some autonomous operations require higher localization accuracy than the one estimated by the ATP system as they are based on the driver's experience.

CV-based algorithms for ego-motion estimation are usually applied in standard datasets such as KITTI [11] and EuRoC-MAV [12]. These datasets are recorded in scenarios with enough illumination and no big light changes. Furthermore, scenarios are composed of images containing relatively sufficient textures and Lambertian surfaces. However, the application of CV algorithms in scenarios with more challenging characteristics is a less researched topic.

For instance, one of the latest benchmark challenges in visually challenging odometry is the Subterranean Challenge (SubT), organized by the Defense Advanced Research Projects Agency (DARPA). Perceptually challenging scenarios and tasks were stated in this challenge, such as navigation through tunnel systems, cave networks, or urban underground environments. The participating teams presented several approaches [13]–[16] to study the robotics autonomy in underground scenarios exploration and navigation. These works emphasize the complexity of localization and navigation in underground environments due to perceptually-degraded conditions. They also emphasize the importance of field testing.

Algorithms applied in urban underground railway scenarios must deal with significant light changes from tunnel areas to platforms, with insufficient illumination and low textures in tunnels. Therefore, considering the cost and the availability of localization systems based on other sensors, the exploration of vision-based localization systems' limitations for an urban underground railway scenario has been pursued. The use of cameras could replace the driver's experience in some autonomous operations.

For that, firstly, a state of the art (SoTA) of ego-motion estimation applied to the railway environment has been made. As the research in the urban underground railway domain is very limited, approaches from other domains such as general robotics and automation have been considered.

However, using ego-motion estimation algorithms in an urban underground railway scenario requires a dataset from this domain. After an analysis of most referenced datasets for ego-motion estimation in the CV community (datasets labeled with 6-DoF pose), no standard dataset from the railway domain was found; hence, a proprietary dataset generation was pursued.

In general, the ground truth of VO datasets is generated using a GPS sensor [11], [17]–[20]. But, the GPS signal is unavailable in underground zones like the urban underground railway domain. Thus, a method that computes the 6-DoF pose of each frame from the train ERTMS/ETCS ATP data, geodetic map coordinates, and railway infrastructure gradient profile was defined and implemented.

Then, an experimental setup was suggested to evaluate the performance of state-of-the-art ego-motion estimation algorithms in the target domain. The validation of the recording setup was done by generating a complementary dataset in an urban driving car domain.

After evaluating the performance of the ego-motion estimation algorithms in the urban underground railway domain and to afford the scenario limitations identified that hinder the use of ego-motion estimation algorithms, the application of a data enhancement technique was proposed. Data enhancement techniques are devoted to dataset transforming to increase the data quality. In this research work, the data enhancement process is dedicated to the lighting limitations of the target domain. It aims to reduce the impact of the severe lighting conditions found in the underground railway scenario.

The main contributions of this thesis are summarized as follows:

- We introduce a ground truth generation method for an ego-motion estimation dataset in the urban underground railway domain. This method is based on the synchronization of camera frames, geodetic coordinates, ERTMS/ETCS ATP train data, and a railway gradient profile. Using the proposed method, we generate a proprietary VO dataset.
- Once having the proprietary dataset, we define an experimental setup for ego-motion estimation evaluation in challenging environments, such as the target scenario. Using standard evaluation metrics and analyzing the experimental results, we measure the performance of the state-of-the-art ego-motion estimation algorithms.
- We propose a GAN-based image enhancement approach to handle the challenges faced by the state-of-the-art ego-motion estimation algorithms in the target scenario. We show that image enhancement improves the performance of the selected algorithms.

This dissertation is organized as follows. Firstly, the SOTA of CV-based ego-motion estimation approaches is carried out in Chapter 2, where the main advantages and drawbacks of localization systems are analyzed. Then, the proposed proprietary dataset generation method is explained. In Chapter 4, the experimental setup for ego-motion estimation in the urban underground railway scenario is explained, and the evaluation of state-of-the-art algorithms is accomplished. Finally, the benefits of data enhancement effect on the state-of-the-art ego-motion estimation algorithms are analyzed in Chapter 5, and the main conclusions of the research are drawn in Chapter 6.

2

VO/vSLAM: SOTA on CV-based ego-motion systems

The CV-based localization research works centered on the autonomous robotics domain started in the late sixties with the Standford Research Institute's work [21]. Later, CV-based localization system development was also extended to new domains, such as, drones or transport vehicles. The research carried out recently on *Intelligent Transportation Systems (ITS)*, *Advanced Driving Assistance Systems (ADAS)*, intelligent infrastructures, and autonomous driving have brought many benefits to the transportation industry [22]. These technologies provide the vehicle with its decision-making capacity and the ability to interpret its environment. Increasing the perceptual vehicle capability through cameras, rather than relying only on the road infrastructure, allows for increased autonomous vehicle context understanding capabilities for localization.

The railway domain is also transforming towards the ITS and the ADAS industry. However, CV-based autonomous train localization research is yet a starting research field.

In this chapter, a SOTA of CV-based localization approaches has been developed. First, as CV-based railway ego-motion systems are being focused on, the application of different purpose CV systems in the target domain are analyzed. The ego-motion can be relative or absolute, depending on the localization system used. Relative or local localization answers to the computation of the position of a robot relative to its start position [23]. Contrarily, absolute or global localization deals with obtaining the absolute position concerning a global reference [24].

The SOTA of CV-based localization approaches starts with introducing the different strategies designed for camera-based robot localization and autonomous navigation: VO and vSLAM. Since only a few research works investigate the use of VO/vSLAM systems in the underground railway, the localization systems developed in robotics and autonomous vehicles applications are analyzed. As the target domain scenario contains some visual or perceptual characteristics that can degrade the performance of the use of state-of-the-art VO/vSLAM algorithms, the literature review also considers VO/vSLAM works in perceptually challenging environments.

2.1 CV systems in the railway domain

The development and evaluation of CV systems in the railway domain is a relatively recent research field. The literature review evidences the limited amount of works published in the railway domain compared to other domains such as robotics, autonomous cars, or UAVs. Figure 2.1 represents a list of perception and decision-making task afforded by the CV based systems with mono or stereo vision cameras in the railway domain.

Perception	Decision making
Localization Obstacle detection Track detection Signal detection and interpretation Platform monitoring	Autonomous driving Track maintenance Track diagnosis Risk assessment

Figure 2.1.: Classification of the main perception and decision making tasks afforded in the railway domain by the CV research community.

The first works, presented in 2010, primarily focused on extracting and presenting information to the driver to assist in train driving and to the operator for infrastructure maintenance. Some of the works have focused on railway track detection [25]–[28] or track anomaly detection [29]–[32] for diagnosis and maintenance purposes.

Other works focused on providing driver support to facilitate driving and increase safety can be found in the literature, such as early detection of intersections, curves, and changes of direction [33], the detection of obstacles on the road as part of a safety system [34]–[36], precise stop operation [37], [38], or the detection and interpretation of signals located in the vicinity of the railway track [39]–[42]. In [43], [44], an approach to estimating the distance to previously detected railway signals is proposed.

Finally, CV systems have also been used for risk assessment [45], vanishing point detection for camera calibration purposes [46], [47] or the monitoring and detection of dangerous situations on the platforms in order to detect objects and people falling onto the tracks and increase safety on them [48].

However, this thesis focuses in the localization task. Concerning the research of CV-based localization systems in the railway domain, the first work was presented by Tschopp *et al.* in 2019 [5]. They presented a camera and IMU-based localization system for rail vehicles. In the last years (2019-2021), train localization systems based on fusing cameras with other sensors such as IMU, Dynamic Vision Sensors (DVS), GNSS, or LIDAR [49]–[54] were published.

2.2 CV-based ego-motion and mapping

The task of retrieving the relative ego-motion and the three-dimensional (3-D) structure of the environment (mapping) from a set of camera images is known in the CV community as *structure from motion* (SfM) [55]. The most addressed strategies in SfM are Visual Simultaneous Location and Mapping (vSLAM) and Visual Odometry (VO).

SLAM was first introduced in [56] stating that the mapping and localization problems are correlated [57], so they can be solved together. SLAM usually refers to a robot equipped with a specific sensor, estimating its motion and reconstructing a model of the surrounding environment [58]. Figure 2.2 represents the different dimensions of mobile robot navigation, and where SLAM is located. When the sensor used in SLAM is a camera, it is called Visual SLAM (vSLAM). It is commonly composed of a VO step where the local trajectory is estimated through consecutive images, and a loop closing and optimization step where a metrically global and consistent map is acquired [55].

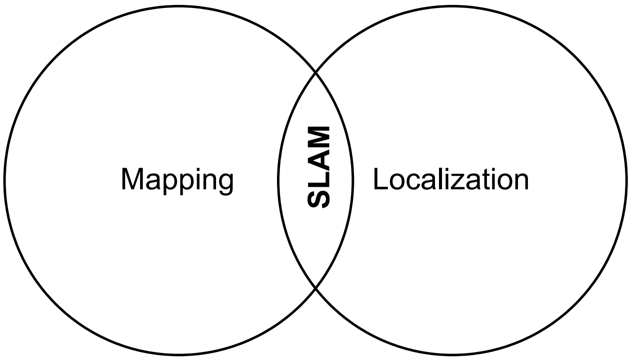


Figure 2.2.: SLAM representation among the different dimensions of mobile robot navigation.

Odometry is a relative localization technique that estimates the robot position change over time starting from an initial point [59], [60]. A robot position is more usually defined as a *pose* that includes the motion translation and the orientation [61], [62].

VO is a particular case of odometry where the position information is acquired through camera images [63]. The term VO was first introduced by Niester *et al.* [64], proposing a method for estimating camera motion using RANSAC [65] outlier refinement method and tracking extracted features across all frames. Before that, feature matching was done just in consecutive frames. Later research works have shown that VO methods might perform as well as wheel odometry while the cost of cameras is much lower compared to more accurate INS- and LiDAR-based systems [63]. Furthermore, VO is one of the most robust techniques used for vehicle localization [60].

VO algorithms usually follow a standard pipeline structure composed of camera calibration, feature extraction and matching, motion estimation, data association, and local optimization. One of the essential steps from this pipeline is feature extraction, which refers to taking relevant data out of an image. Depending on the task, relevant data can represent different features: corners, edges, or even more complex objects. Feature matching is the process of finding a correspondence between the extracted features across different frames.

The relation between SfM, VO, and vSLAM is shown in figure 2.3. SfM is a broader concept than vSLAM and VO, as it focuses on 3-D reconstruction of the environment and is usually performed in unordered and distinct image sequences. VO can be considered the primary process of vSLAM; however, vSLAM includes other processes such as loop closure or global optimization. While VO focuses on estimating the relative motion from a moving camera, vSLAM comprises other components for 3-D reconstruction of the environment for global and consistent motion estimation.

The selection of VO and vSLAM algorithms depends on the trade-off between performance and consistency. VO algorithms are faster and simpler to implement, while vSLAM achieves potentially more precise results but trades off performance. VO algorithms accumulate a drift in the trajectory estimation as they compute the motion incrementally, and the errors introduced by each new measure are combined over time. However, this drift can be corrected through local optimization or a combination with other sensors [55], [66]–[70].

Depending on the selected camera, VO/vSLAM algorithms can be considered monocular (one lens) or stereo (two or more lenses). Usually, all the VO/vSLAM algorithms

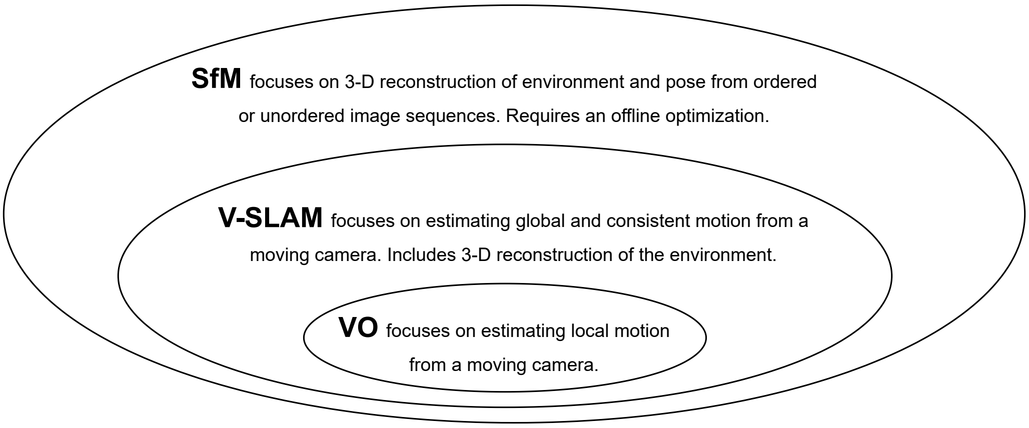

SfM focuses on 3-D reconstruction of environment and pose from ordered or unordered image sequences. Requires an offline optimization.
V-SLAM focuses on estimating global and consistent motion from a moving camera. Includes 3-D reconstruction of the environment.
VO focuses on estimating local motion from a moving camera.

Figure 2.3.: Relation of different CV-based localization strategies: SfM, vSLAM and VO.

are classified in one or the other category. However, some algorithms can be configured for both camera modes. Additionally, some algorithms are considered monocular even if they include some preprocessing steps with stereo images (e.g., training deep networks) if they run with monocular images (e.g., [71],[72] or [73]). Most VO/vSLAM research works have focused on monocular algorithms where the aim is to estimate the motion information from consecutive images. However, lately, the interest in stereo approaches has increased in the VO/vSLAM community, primarily by including stereo information into monocular pipelines.

Depending on the strategy used to estimate ego-motion, VO/vSLAM algorithms can be classified as *learning-based* and *geometry-based* [74]. Figure 2.4 shows the general classification of VO/vSLAM algorithms. First, geometry-based VO/vSLAM algorithms appeared, aiming to estimate the motion through the geometric characteristics of images. Later, learning-based VO/vSLAM algorithms were targeted as they showed great potential in other research fields. The learning-based algorithms research increased considerably with the advances of Deep Learning and their application to this research domain.

Geometry-based VO/vSLAM approaches rely on image geometric characteristics and camera model to reconstruct the ego-motion between consecutive frames. *Learning-based VO/vSLAM* algorithms can estimate the pose directly from an input image without feature extraction or feature matching processes. Compared with geometry-based algorithms, learning-based algorithms are more robust to changing environments with specific conditions as they can learn more effective feature representations [75].

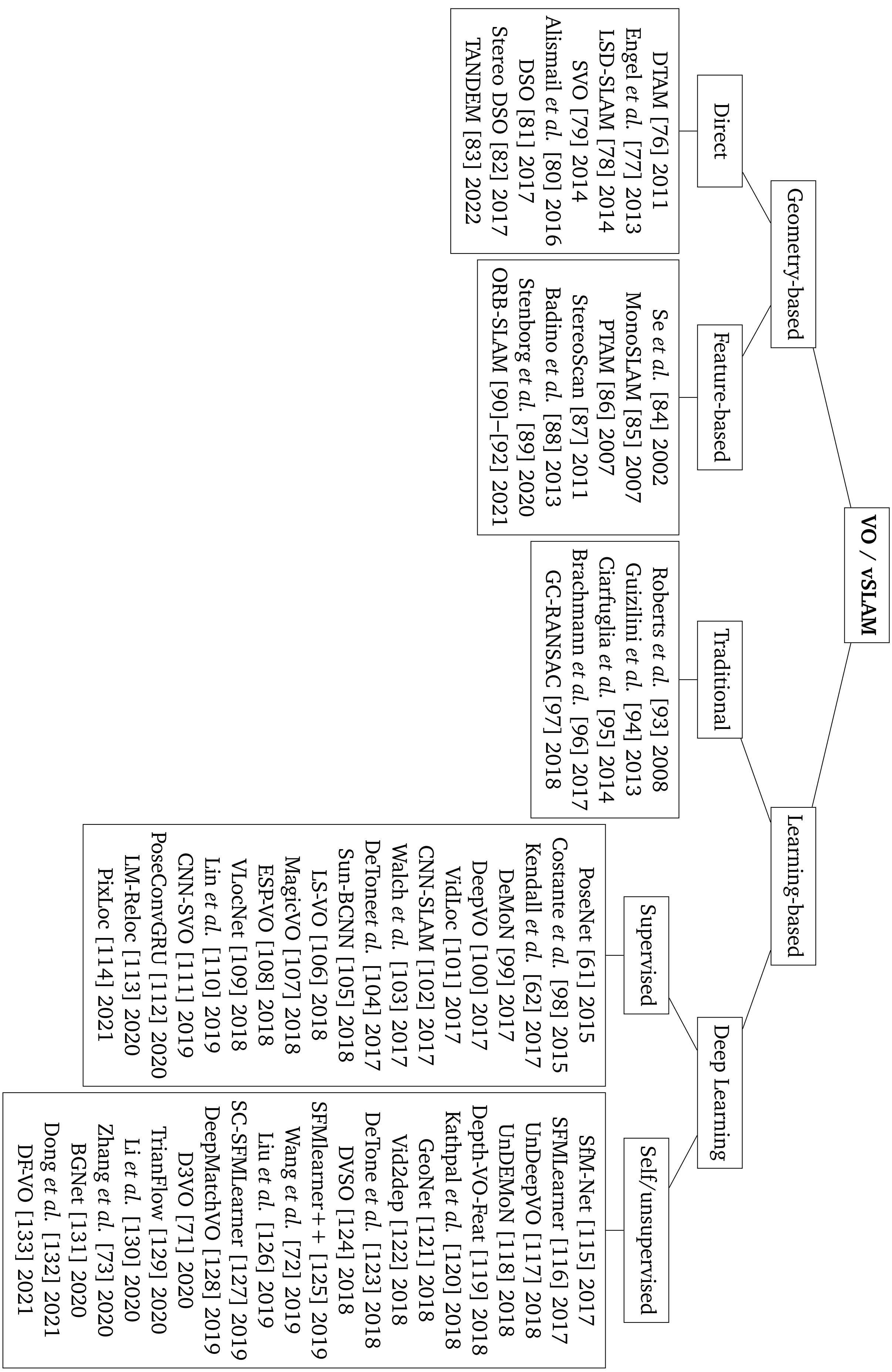


Figure 2.4.: General classification of VO/vSLAM algorithms based on the strategy used to estimate ego-motion and reconstruct the environment.

2.2.1 Geometry-based VO/vSLAM

Geometry-based VO/vSLAM is usually divided into feature-based VO/vSLAM and direct VO/vSLAM (also known as appearance-based). Table 2.1 resumes the most relevant geometry-based VO/vSLAM algorithms. The feature-based algorithms rely on image features that were first studied by Harris *et al.*, in the late 1980s, by exploring corner and edge detectors [134]. Based on Harris' work, feature-based ego-motion estimation algorithms arose. The first feature-based VO/vSLAM algorithms utilized artificial landmarks as visual patterns for localization [135]. In the early 2000s, approaches that use natural visual landmarks appeared with the advances in CV application. The first feature-based algorithms with natural landmarks aimed the robot relocalization in unknown environments. Later, feature-based works started to focus on SLAM and 3D reconstruction. At the same time, direct methods appeared aiming to use all the dense data in an image to estimate the ego-motion of a moving robot.

Feature-based VO/vSLAM

Feature-based methods rely on image feature detection and matching. They have good accuracy, are robust in dynamic scenes, and deal with viewpoints variances [136]. However, unlike direct methods, feature-based techniques can be inadequate in low texture areas. They are optimized for execution speed rather than pose estimation precision [79]. Usually, feature-based methods follow a standard pipeline: first, image features are extracted, and then the features are matched in successive frames to recover the camera motion and environment structure. Finally, the obtained pose estimation and structure are refined and optimized by minimizing the re-projection error.

Several feature extraction and matching methods can be found in the literature, and usually, they are combined. The feature extraction and matching strategies were first designed to detect specific image key-points (e.g., SIFT [137] or FAST [138]), and with the advances in Machine Learning, strategies for learning feature extraction and matching approaches appeared.

The first relevant feature-based VO/vSLAM algorithm appeared in 2002; Se *et al.* [84] described a vision-based mobile robot localization and mapping algorithm that uses SIFT features as natural visual landmarks for a feature matching process in dynamic environments. Davison *et al.* [85] presented MonoSLAM in 2007, a localization and mapping algorithm that creates a feature map based on a probability

	Algorithm	Year	Mode	Characteristics
Feature-based	Se *et al.* [84]	2002	S	Robot relocalization, SIFT features as landmarks
	MonoSLAM [85]	2007	M	SLAM, probability framework-based features
	PTAM [86]	2007	S	Tracking and mapping, Separate threads for tracking and mapping
	StereoScan [87]	2011	S	3-D reconstruction, sparse feature matching into a VO pipeline
	Badino *et al.* [88]	2013	S	VO, multi-frame features and optical flow and stereo disparity for reprojection error minimization
	Stenborg *et al.* [89]	2020	M	Long-term localization, image-sequence based
	ORB-SLAM [90]–[92]	2015-2017-2021	M+S	SLAM, ORB features, loop closure and Bundle Adjustment
Direct	DTAM [76]	2011	M	Ego-motion estimation, dense 3-D surface model
	Engel *et al.* [77]	2013	M	Semi-dense map
	LSD-SLAM [78]	2014	M	SLAM, large-scale map estimation
	SVO [79]	2014	M	Ego-motion estimation, probabilistic mapping method
	Alismail *et al.* [80]	2016	S	Feature descriptor alignment for VO
	DSO [81]	2016	M	
	Stereo DSO [82]	2017	S	Ego-motion estimation, probabilistic model for photometric error minimization, stereo constraints
	TANDEM [83]	2022	M	SLAM, deep multi-view stereo

* M=Monocular, S=Stereo, M+S=Monocular and Stereo, M(S)=Monocular with Stereo training

Table 2.1.: Geometry-based VO/vSLAM algorithms.

framework and combines it with a general camera motion model and feature initialization. However, localization and mapping are done in a single thread, and, therefore, a limited number of features can be handled. To increase the number of features, Klein *et al.* proposed Parallel tracking and mapping (PTAM) [86], with separate tracking and mapping threads to use more computationally expensive operations. This algorithm showed successful results in a small indoor environment.

Later, in 2011, Geiger *et al.* presented StereoScan [87]. This 3-D reconstruction algorithm is based on an efficient and robust VO pipeline with a sparse feature matching process. In 2013, Badino *et al.* [88] proposed a VO approach that computes optical flow and stereo disparity to minimize the reprojection error of the tracked features using a multi-frame feature integration strategy. In [139], Vakhitov *et al.* presented a stereo relative pose estimation algorithm using a minimal set of line and point features.

In 2015, ORB-SLAM was presented by Mur-Artal *et al.* [90]. ORB-SLAM is an efficient and accurate geometric localization approach that has become the state-of-the-art comparison for all the later methods. It is based on the ORB [140] feature matching approach, a bundle adjustment algorithm, and a loop closure strategy. Later, in 2017, it was extended as ORB-SLAM2 [91], including stereo and RGB-D cameras, and, in 2021, as ORB-SLAM3 [92] by incorporating inertial sensors and multimap SLAM. Figure 2.5 depicts the ORB-SLAM2 architecture.

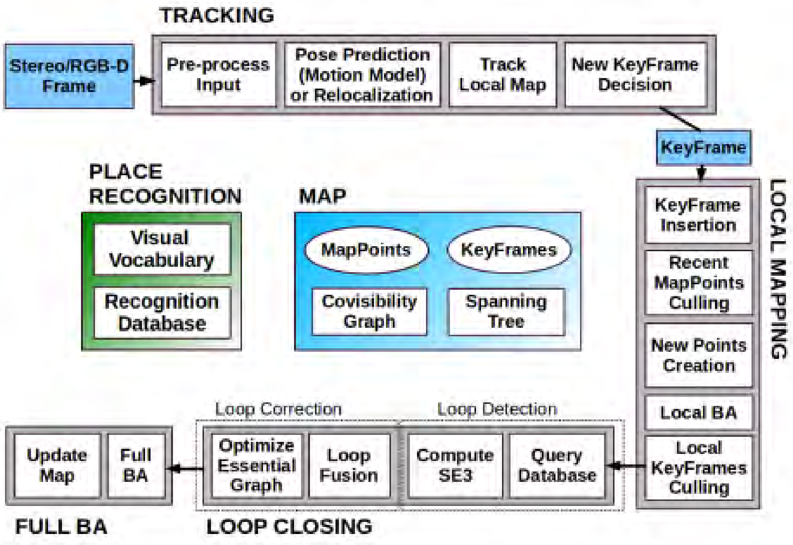


Figure 2.5.: ORB-SLAM2 [91] architecture. It is based on feature tracking and mapping processes based on ORB features. Then a loop closure and Bundle Adjustment strategies are used for global optimization.

One of the research lines in the feature-based algorithm usage is the investigation of robust feature detectors and descriptors, allowing feature-based algorithms in several scenarios and domains. Aiming to increase the robustness of feature detectors and descriptors, many research works dedicated to improving these feature extraction techniques have been published: by introducing learning methods (explained in Section 2.2.2) or exploiting spatial matching criteria. One of the most referenced methods is the ORB feature extractor [140]. ORB is based on pixel brightness to extract FAST keypoints and then compute BRIEF features.

Direct VO/vSLAM

On the contrary, direct VO/vSLAM techniques operate directly on intensity values estimating ego-motion by minimizing photometric error. One of the first relevant VO/vSLAM algorithms is Dense Tracking and Mapping (DTAM), presented by Newcombe *et al.* [76] in 2011. A direct VO/vSLAM ego-estimation method that creates a dense 3-D surface model using the whole image instead of extracting features. However, an external depth map initialization is required. Later, in 2014, Engel *et al.* [78] proposed a semi-dense Large-Scale Direct monocular SLAM (LSD-SLAM) method, which tracks the camera motion and creates a large-scale map of the environment. This approach was based on their previous work on semi-dense depth estimation [77], where semi-dense refers to the map being dense just in image regions that carry information.

Semi-direct visual odometry (SVO) [79] is one of the most predominant approaches among direct VO/vSLAM algorithms. It uses a probabilistic mapping method to estimate ego-motion and explicitly models outlier measurements. In 2017, Wang *et al.* presented Stereo Direct Sparse Odometry (Stereo DSO) [82], a method for VO estimation from stereo cameras based on the previously proposed monocular DSO algorithm [81]. It is based on a probabilistic model for photometric error minimization and an optimization process for pose estimation. They also introduced constraints from stereo cameras to the bundle adjustment pipeline. Lately, Koestler *et al.* presented TANDEM [83], a SLAM system that estimates ego-motion based on a direct VO pipeline and deep multi-view stereo.

However, the performance of direct VO/vSLAM algorithms degrades if the dataset is not photometrically calibrated. In addition, it is sensitive to geometric distortions such as those induced by the camera speed [82]. Furthermore, as mentioned in [80], direct methods require a constant irradiation appearance between matched pixels.

In general, geometry-based approaches are efficient when there is enough illumination and texture to match features among different frames, sufficient overlap between frames and scene is static [5]. However, geometric VO/vSLAM works suffer from scale drift issues where scale is inconsistent and expensive global bundle adjustment algorithms are applied to minimize the problem. Furthermore, they do not have the adaptive capabilities to readjust to scenarios in peculiar circumstances. Monocular VO/vSLAM algorithms have a depth-translation scale ambiguity issue [141] that makes the estimations up to scale. When using more than one camera, the translation magnitude can be extracted from triangulation and depth information [142]. However, one dimension is missing using only one camera, and the scale cannot be directly measured.

2.2.2 Learning-based VO/vSLAM

With the increase of computational resources, efficiency, and Deep Learning algorithms improvement, the use of Machine Learning approaches in Computer Vision problems has grown, becoming the dominant approach nowadays. This situation also comes from the promising results obtained by the application of deep learning techniques in other computer vision tasks, specifically with the use of Convolutional Neural Networks (CNN) on large-scale image classification (Krizhevsky *et al.* [143]) and the development of Recurrent Neural Networks (RNN) on Natural Language Processing research. As learning-based VO/vSLAM approaches are based on Machine Learning, and mainly on deep learning techniques, the research on learning-based VO/vSLAM algorithms has also grown in the last few years. Learning-based approaches can solve the scale-ambiguity issue in some VO/vSLAM algorithms as their predictions are associated with a real-world scale.

The first learning-based methods used optical flow information for ego-motion estimation through different traditional strategies, such as K Nearest Neighbor (KNN) in a moving robot (Roberts *et al.* [93]), multiple-output Gaussian process (MOGP) (Guizilini *et al.* [94]), or Support Vector Machines (SVM) (Ciarfuglia *et al.* [95]). Later research works have tried to adapt traditional non-learning approaches in Deep Learning pipelines. In this context, RANSAC algorithm has been improved by several works, such as in [96], [97]. More specifically, Brachmann *et al.* proposed Differentiable Sample Consensus (DSAC) [96] algorithm based on RANSAC [65]. They applied DSAC for a camera localization problem, learning an end-to-end camera localization pipeline. Barath *et al.* presented Graph-Cut RANSAC (GC-RANSAC) [97], which improves local optimization using a graph-cut algorithm.

Learning-based VO/vSLAM algorithms rely on learning parts of a standard VO/vSLAM pipeline or designing end-to-end trainable algorithms for ego-motion estimation. As in the case of geometric algorithms, a local optimization step can be added to learning-based algorithms.

The learning-based algorithms studied in this research work can be considered *supervised* or *self/unsupervised*, depending on the network learning approach. Supervised algorithms learn by induction from known examples, usually named ground truth. These known examples are compared to the networks' estimations obtaining an error usually characterized by a loss function [144]. The learning process minimizes this error by updating the networks' weights in training. However, supervised learning requires a considerable amount of labeled data for the training process, which can be costly in terms of time or resources.

In contrast, self/unsupervised algorithms do not require known examples from which to learn. The strategy of self/unsupervised algorithms is to discover the underlying properties and patterns of the data. Usually, unsupervised algorithms are used in clustering, association, and dimensionality reduction tasks, while self-supervised are more focused on regression and classification tasks [145]. Self-supervised algorithms operate in the same way as supervised algorithms. However, they do not require labeled data. They can use their output as a supervision signal in the training process.

Learning features

Some previously mentioned feature-based algorithms have been improved by learning feature extraction, description, and/or matching steps. GCNv2 was presented by Tang *et al.* [146], a deep learning-based feature generation network that can replace ORB-like feature extraction algorithms. Dusmanu *et al.* [147] proposed a cross-descriptor SLAM approach that leverages the descriptor type, which increases its effectiveness. Cavalli *et al.* introduced AdaLAM [148], an outlier detection filter able to be introduced in any VO/vSLAM pipeline to make the algorithm more robust to different domains. However, most of these methods cannot capture the global context and require a substantial aggregation of hundreds of points to predict a pose [62].

Some of the works have also tried to deal with scenario challenges, as Costante *et al.* [98]. They presented a supervised learning approach to learning feature extraction and Frame to Frame (F2F) motion estimation from CNNs in scenarios with blur, luminance, and contrast anomalies. DeTone *et al.* [104] proposed a tracking system using two CNNs, one as a feature detector and the second to calculate the homography between pairs of point images gathered from previous CNN. Later, they presented a self-supervised learning framework [123] to compute image features and use them in a VO pipeline. The results show that learning the best features for the VO frontend improves VO backend results.

Yu *et al.* [131] presented BGNet, a hierarchical unsupervised visual localization method to learn 2D-3D matching from a bipartite graph network. Lately, LM-Reloc was presented by Sumberg *et al.* [113], a relocalization approach based on direct image alignment that uses two deep networks to learn dense visual descriptors (LM-Net) and initialize image alignment (CorrPoseNet). Feature learning can also be improved through ego-motion data incorporation to feature learning algorithms, such as in [149].

Learning depth and flow estimation

Camera-based VO/vSLAM algorithms are based on images where the environment is captured in two dimensions. The third dimension is lost and, therefore, so does the depth information of the environment. However, depth information is crucial for the localization as it enables the inference of the scene geometry from 2D images. Moreover, it allows scale recovery [150] and the distinction of foreground and background points, allowing a better environment understanding. The depth information is used by most recent learning-based VO/vSLAM algorithms in the pose estimation process (e.g., [102], [111]).

Together with depth estimation, the optical flow estimation is also a critical component of some VO/vSLAM algorithms (e.g., [106]). Flow estimation refers to the computation of pixel-wise motions between consecutive images [151]. Usually, the flow estimation describes a vector field where each pixel position gets a displacement vector [152]. This information is used to model the motion between consecutive images.

Some recent works, make use of both depth estimation and flow estimation in the pose estimation process (e.g., [121], [133]). The research works from the literature emphasize the importance of an accurate depth and flow estimation for VO/vSLAM. Therefore, the study of learning depth and/or flow has become an essential research direction for the VO/vSLAM research community.

Supervised VO/vSLAM

One of the first and most relevant supervised learning algorithm is PoseNet, proposed by Kendall *et al.* [61] in 2015, a robust and real-time monocular relocalization system based on an end-to-end trained CNN. This approach was later improved by adding a fundamental treatment of scene geometry introducing geometric loss functions [62]. It is not the unique work focused on loss functions when trying to improve ego-motion estimation. Valada *et al.* presented VLocNet [109], a CNN-based algorithm for 6-DoF global pose regression from a sequence of monocular images. It is a supervised deep learning approach, end-to-end trainable, and with an auxiliary learning loss (geometric consistency loss) to correct relative poses.

Although most learning-based works rely on CNNs to extract features from images and compute poses, the use of Recurrent Neural Networks (RNN) has increased lately. RNNs introduce temporal modeling capabilities to the algorithms by using internal memory. One of the most popular RNN networks introduced in the learning-based VO/vSLAM algorithms is the Long Term Shot Memory (LSTM) [153]. The combination of CNNs and RNNs is called RCNNs, and it benefits from the advantages of both networks, the feature extraction capabilities of the CNN, and the sequential modeling of the RCNN.

The first works that used RCNNs for pose estimation appeared around 2017. Wang *et al.* presented DeepVO [100], an approach that infers camera poses directly on an end-to-end manner from a sequence of RGB frames. They show that end-to-end Deep Learning approaches can be a viable complement to the traditional VO algorithms. After that, in 2018, they presented UnDeepVO [117], an updated version of their previous algorithm where an unsupervised deep learning scheme does the 6-DoF pose estimation, and the absolute scale can be recovered from stereo images.

Clark *et al.* [101] proposed VidLoc, a deep spatio-temporal model for global localization from a monocular image sequence. They introduced RNNs to exploit temporal dependencies and improve the monocular image sequence localization accuracy. Walch *et al.* presented camera pose regression algorithms based on RCNNs to reduce the dimensionality of deep learning module output [154].

In 2018, Jiao *et al.* presented MagicVO [107], a 6-DoF absolute-scale pose estimator framework based on CNNs and Bi-directional LSTM (Bi-LSTM) trained in an end-to-end manner from continuous monocular images. Richer features than with other architectures are extracted increasing the number of filters in the CNN, while the geometric relationship between image sequences is learned from Bi-LSTM.

In most works, pose estimation is incremental, and therefore, the errors can increase substantially in scenarios with long sequences. Some works are devoted to reducing the estimation errors of VO/vSLAM algorithms in long-length scenarios. For example, Lin *et al.* presented a deep end-to-end algorithm based on RCNNs for the long-term 6-DoF VO task [110]. Global and relative sub-networks are implemented to avoid scale drift issues and smooth the VO trajectory. Later, Zhai *et al.* proposed PoseConvGRU [112]. This two-module Long-term RCNN algorithm estimates the ego-motion of an image sequence through a feature encoding module and a memory module.

Although most learning-based pose estimation networks have been based on CNNs or RCNNs, other approaches have included Bayesian CNNs (BCNNs), such as Peretroukhin *et al.* [105]. They presented Sun-BCNN, an approach to introduce global orientation information from the sun into a VO pipeline from an image sequence based on BCNNs. They also estimate the uncertainty, a strategy that some works have also pursued. Following this idea of estimating uncertainty and the pose from a sequence of monocular images, Wang *et al.* proposed ESP-VO [108], which uses RCNNs.

One of the latest approaches that follows this same direction is D3VO [71]. A self-supervised framework that infers camera pose, depth, and uncertainty altogether. The deep models are trained using stereo images and then are incorporated into a direct VO pipeline.

As stated previously, most VO/vSLAM networks utilize flow or/and depth estimation for pose regression. Following this research line, Costante *et al.* presented Latent Space VO (LS-VO) [106], a deep learning-based approach consisting of an autoencoder network to predict optical flow followed by a pose estimation network trained in an end-to-end manner. In the same way, Ummenhofer *et al.* proposed DeMoN [99], a learning-based VO approach that computes depth and camera motion at the same time using encoder-decoder network-based architecture.

Tateno *et al.* [102] proposed CNN-SLAM, a method where CNN-predicted dense depth maps are fused with depth measurements obtained from direct monocular SLAM. This fusion improves localization in low-textured regions and scales recovering where direct SLAM methods fail. Based on SVO [79], Loo *et al.* introduced CNN-SVO [111], an improved VO approach that uses a depth prediction neural network when initializing the map point.

Other research lines have also been proposed in addition to the previously mentioned approaches. Sarlin *et al.* proposed PixLoc [114], a camera localization approach that estimates a 6-DoF pose from an image and a 3D model. Although it sows great generalization ability, it requires a 3D scene representation, which is not always available in some environments and scenarios.

	Algorithm	Year	Mode	Characteristics
Supervised	PoseNet [61]	2015	M	Relocalization, CNN, end-to-end
	Costante *et al.* [98]	2015	S	Pose estimation, feature extraction, F2F
	Kendall *et al.* [62]	2017	M	Pose estimation, CNN, geometric loss function
	DeMoN [99]	2017	M	Pose estimation, depth and motion through encoder-decoder
	DeepVO [100]	2017	M	Pose estimation, RCNN, outdoor and indoor
	VidLoc [101]	2017	M	Global localization, RCNN
	Walch *et al.* [103]	2017	M	Pose regression, RCNN
	CNN-SLAM [102]	2017	M	SLAM, CNN, depth
	DeTone *et al.* [104]	2017	M	Tracking, CNN, feature learning
	Sun-BCNN [105]	2018	M	Pose estimation, BCNN, uncertainty
	LS-VO [106]	2018	M	Pose estimation, optical flow, autoencoder, end-to-end
	MagicVO [107]	2018	M	Pose estimation, CNN, Bi-LSTM, end-to-end
	ESP-VO [108]	2018	M	Pose estimation, RCNN, uncertainty
	VLocNet [109]	2018	M	Pose regression, CNN, end-to-end, geometry consistency loss
	Lin *et al.* [110]	2019	M	Pose estimation, RCNN, long-term, end-to-end
	CNN-SVO [111]	2019	M	Pose estimation, CNN, depth
	PoseConvGRU [112]	2020	M	Pose estimation, RCNN
	LM-Reloc [113]	2020	S	Relocalization, image alignment, descriptor learning
	PixLoc [114]	2021	M	Relocalization, 3D representation

* M=Monocular, S=Stereo, M+S=Monocular and Stereo, M(S)=Monocular with Stereo training

Table 2.2.: Supervised learning-based VO/vSLAM algorithms.

Unsupervised VO/vSLAM

Scene depth and flow estimation are crucial for most learning-based VO/vSLAM algorithms, and many self/unsupervised approaches have tried to deal with them. In 2017, SfM-Net [115] was presented, a learning-based motion estimation approach by estimating scene depth, camera motion, and 3D object rotations and translations.

Later, UnDEMoN was presented by Babu *et al.* [118], an unsupervised learning VO system for depth and pose estimation through an objective function and temporally aligned image sequences.

One of the most important unsupervised VO/vSLAM works is SFMLearner [155], a system for deep tracking the camera using CNNs in an end-to-end learning approach and single-view depth estimation. Later, it has been improved by several works, as in [120] by introducing depth estimation from multiple views, or in [125] by updating the loss function introducing the Epipolar constraints.

As this last work, some of the self/unsupervised VO/vSLAM works have focused on loss function that optimizes the depth, flow or pose estimation. In [119], Zhan *et al* proposed Depth-VO-Feat with a feature reconstruction loss for depth and VO estimation without scale ambiguity trained with stereo video sequences. However, no occlusion assumption is made, and the scene must be rigid.

Other works proposed loss functions to handle challenging scenario characteristics. Yin *et al.* proposed GeoNet [121], an unsupervised learning framework for depth, optical flow, and ego-motion estimation from image sequences. They introduced an adaptive geometric consistency loss to increase robustness towards outliers and non-Lambertian surfaces.

However, GeoNet is not the only algorithm centered on the reflectivity of the surfaces. Mahjourian *et al.* [122] proposed vid2dep, an unsupervised learning algorithm to estimate depth and ego-motion from a monocular image sequence considering the geometric constraints of the environment by introducing a 3D-based loss. Also, in [128], Shen *et al.* presented DeepMatchVO, a self-supervised monocular approach for VO. They introduced the matching loss that includes the photometric and geometric losses to avoid significant systematic errors due to occlusions and reflective surfaces.

As previously mentioned, monocular VO/vSLAM algorithms suffer from a scale inconsistency due to the inability to estimate the real depth-translation scale from a single-lens camera. Trying to handle this scale ambiguity, SC-SFMLearner [127] was proposed. It is based on a geometric consistency loss to solve the scale ambiguity over the frames and a self-discovered mask to handle moving objects and occlusions.

As with supervised algorithms, some of the unsupervised VO/vSLAM methods have approached the ego-motion estimation by incorporating RCNNs. Liu *et al.* [126] presented an unsupervised end-to-end trainable algorithm to estimate monocular VO based on RCNNs and an absolute scale recovery method. In [130], Li *et al.*

proposed an online self-supervised VO algorithm that estimates pose and adapts to new environments using LSTMs.

Other approaches have also tried to design algorithms adaptable to dynamic environments. Dong *et al.* [132] proposed a camera relocalization algorithm based on outlier-aware neural trees for dynamic indoor environments.

Finally, some recent approaches have tried to handle ego-motion estimation by incorporating geometric strategies into learning-based algorithms. Wang *et al.* [72] proposed an unsupervised ego-motion estimation algorithm based on CNNs, the Kalman filter introduction into the learning framework, and an encoder-decoder architecture for depth estimation. In [129], Zhao *et al.* proposed TrianFlow, a self-supervised depth and pose learning framework with a robust scale recovery method based on fundamental matrix solving, triangulation, and depth reprojection error estimation. Zhang *et al.* [73] presented an unsupervised method for depth and stereo camera motion estimation through a depth consistency loss based on the triangulation principle.

One of the of the most promising unsupervised works is DF-VO [133], presented by Zhan *et al.*. It outperforms pure deep learning-based and geometry-based methods and solves the scale-drift issue by adding a scale consistent single-view depth CNN. For that, first, it recovers high-quality correspondences from deep flows obtained from LiteFlowNet [156]. The correspondence selection is computed through forward-backwards flow consistency. Two alternative trackers do the pose estimation depending on the previously found correspondences (E-tracker or PnP-tracker). Finally, it handles the scale drift issue by comparing triangulated depths and deep depths obtained by a depth network based on Monodepth2 [157]. Architecture of DF-VO is depicted in Figure 2.6.

Apart from supervised and /self/unsupervised approaches, semi-supervised VO/vSLAM approaches can also be found in the literature. Semi-supervised learning-based algorithms are trained through labeled and unlabeled data [158]. One of the most referenced semi-supervised works is DVSO [124]. It leverages deep monocular depth prediction to overcome the limitations of geometry-based VO.

2.3 VO/vSLAM in visually degraded environments

In general, VO/vSLAM algorithms have been intensely studied in the CV and robotics communities; however, most research works have focused in the same standard

	Algorithm	Year	Mode	Characteristics
Self/unsupervised	SfM-Net [115]	2017	M	Pose and depth estimation, 3D object motion
	SFMLearner [116]	2017	M	Deep tracking, CNN, end-to-end
	UnDeepVO [117]	2018	M	Pose estimation, scale recovery, end-to-end
	UnDEMoN [118]	2018	S	Pose and depth estimation, temporal alignment
	Depth-VO-Feat [119]	2018	S	Pose and depth estimation, stereo, reconstruction loss, rigid scene
	Kathpal *et al.* [120]	2018	M	Pose and depth estimation, multiple views, SSIM, data augmentation
	GeoNet [121]	2018	M	Pose, depth and optical flow estimation, adaptive geometry consistency loss
	Vid2dep [122]	2018	M	Pose and depth estimation, 3D-based loss
	DeTone *et al.* [123]	2018	M	Feature learning for pose estimation
	DVSO [124]	2018	S	Pose and depth estimation, stereo
	SFMlearner++ [125]	2019	M	Pose estimation, Epipolar constraints
	Wang *et al.* [72]	2019	M(S)	Pose and depth estimation, CNN, encoder-decoder architecture
	Liu *et al.* [126]	2019	M	Pose estimation, RCNN, absolute scale recovery
	SC-SFMLearner [127]	2019	M	Pose estimation, mask for dynamic environments, long sequences
	DeepMatchVO [128]	2019	M	Pose estimation, matching loss
	D3VO [71]	2020	M(S)	Pose, depth and uncertainty estimation, stereo
	TrianFlow [129]	2020	M	Pose and depth estimation, triangulation
	Li *et al.* [130]	2020	M	Pose estimation, LSTM, adaptive to new environments
	Zhang *et al.* [73]	2020	M(S)	Pose and depth estimation, CNN, stereo
	BGNet [131]	2020	M	Pose estimation, 2D-3D matching learning
	Dong *et al.* [132]	2021	M	Pose estimation, neural trees, dynamic environments
	DF-VO [133]	2021	M+S	Pose, depth and optical flow, estimation, CNN, E-tracker, PnP

* M=Monocular, S=Stereo, M+S=Monocular and Stereo, M(S)=Monocular with Stereo training

Table 2.3.: Self/unsupervised learning-based VO/vSLAM algorithms.

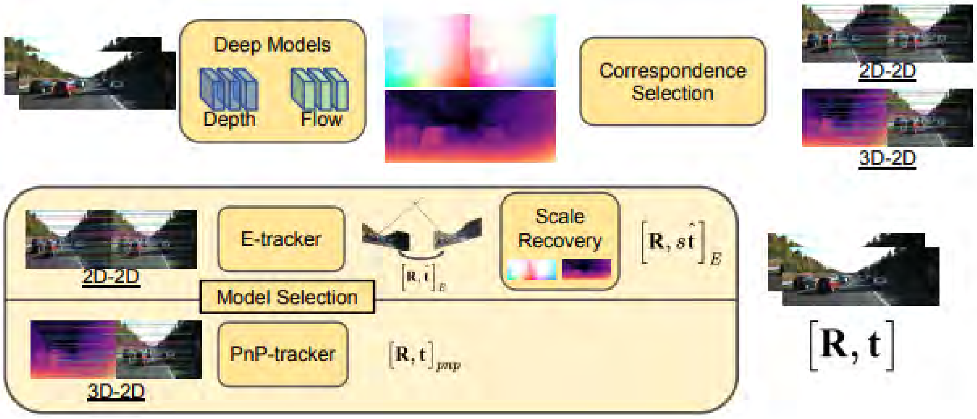

Deep Models
Depth
Flow
Correspondence Selection
2D-2D
3D-2D
2D-2D
E-tracker
$[\mathbf{R}, \mathbf{t}]_E$
Model Selection
Scale Recovery
$[\mathbf{R}, s\hat{\mathbf{t}}]_E$
3D-2D
PnP-tracker
$[\mathbf{R}, \mathbf{t}]_{pnp}$
$[\mathbf{R}, \mathbf{t}]$

Figure 2.6.: DF-VO [133] architecture. For the depth deep model it uses Monodepth2 [157] algorithms and LiteFlowNet [156] for the flow estimation. Depending on the flow consistency computed by LiteFlowNet, a tracker is selected that will estimate a 6-DoF pose through the E-tracker (2D-2D correspondences) or the PnP-tracker (3D-2D correspondences).

scenarios and more study on complex environments is needed [159]. VO/vSLAM algorithms aiming to derive localization data through visual sensors are usually evaluated and compared by reference standard datasets such as KITTI [11], [160] (e.g., [71], [100], [116], [119], [124], [133]are evaluated in the KITTI dataset). These datasets are mainly recorded in outdoor scenarios and are composed of images containing relatively sufficient textures and Lambertian surfaces. However, few algorithms, datasets, and benchmarks can be found in challenging scenarios with varying light conditions, low illumination, low textures, or non-Lambertian surfaces. E.g.,

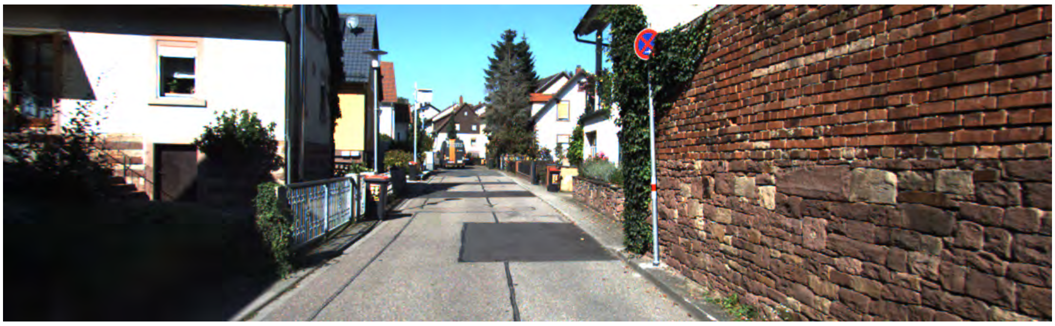


Figure 2.7.: Sample image from the KITTI dataset where visual characteristics of the recorded scenarios are shown. A lot of textures can be appreciated in all the image regions and with good lighting conditions.

As mentioned in [60], literature VO solutions have limitations in challenging scenarios that contain insufficient illumination and textures or variable lighting conditions. Literature VO solutions require sufficient illumination and enough textured surfaces in the environment for correct feature matching. A good illumination allows motion extraction from images, as pixel displacement can be inaccurately estimated otherwise.

For instance, one of the latest benchmark challenges in visually challenging odometry is the Subterranean Challenge (SubT), organized by the Defense Advanced Research Projects Agency (DARPA) in 2018. Perceptually challenging scenarios and tasks were stated in this challenge, such as navigation through tunnel systems, cave networks, or urban underground environments. The participating teams presented several approaches [13]–[16] to study the robotics autonomy in underground scenarios exploration and navigation. These works emphasize the complexity of localization and navigation in underground environments due to their perceptually-degraded conditions. They also emphasize the importance of field testing.

Other research works that have explored the difficulties derived from the environment characteristics for VO/vSLAM can also be found in literature: [89] focuses on scenarios under day-night or seasonal changes; in [131] BGNet is proposed, an algorithm that handles scenes with illumination changes or repetitive patterns; [121] proposes GeoNet, a robust algorithm for non-Lambertian surfaces; or, low-textured regions are explored in CNN-SLAM [102].

However, as stated before, VO/vSLAM application in challenging scenarios such as the urban underground railway domain is a less researched topic. These scenarios are characterized by some conditions that can hinder the use of state-of-the-art VO algorithms. As Almalioglu *et al.* [161] point out, these techniques usually rely on finding correspondences between consecutive frames, requiring certain environmental conditions such as adequate lighting conditions (good illumination, similar lighting conditions in subsequent frames), sufficient textures, and Lambertian surfaces. Besides, cameras tuned to work in both lighting conditions give blurring and noisy images.

2.4 Localization sensors for ego-motion estimation through sensor fusion

CV-based localization systems have been shown to be robust, reliable, and provide meaningful information [162]. However, image processing algorithms tend to be computationally expensive and are deeply sensitive to circumstantial conditions like lighting, shadows, textures, blurring, or surfaces characteristics.

The research on VO/vSLAM works in those challenging environments has pushed the exploration of sensor fusion for CV-based ego-motion estimation. The VO/vSLAM robotics community has also pointed out this research direction lately [15], [16]. Some recent works show that the addition of data from non-visual sensors to the information provided by the cameras can improve the robustness and/or accuracy of CV-based localization algorithms [163]. This section contains a review of different sensors used in robot localization research.

Various sensors for mobile robot positioning, such as wheel sensors, cameras, laser sensors, Global Positioning System (GPS), Global Navigation Satellite System (GNSS), and Inertial Navigation System (INS), have been presented in the last few years.

Wheel sensors are one of the most popular sensors used in mobile robot localization systems [59] and are mainly based on wheel encoders that measure the rotation of the wheels. However, wheel encoders suffer from drift issues due to wheel slippage [164].

Triangulation and propagation time measurements derive the global location in GNSS-based localization systems. The GPS is the most widely used localization sensor because it provides absolute position without relative error accumulation. Nevertheless, it cannot be used in indoor scenarios [59] as it is limited to satellite signal availability and suffers in constraining environments. Furthermore, when using standard GNSS alone, the localization accuracy is not enough for some autonomous operations requiring a high accuracy [165].

INS is a relative localization system based on an Inertial Measure Unit (IMU) to estimate position and velocity from an initial point. The IMUs are typically comprised of gyroscopes and accelerometers. However, like the other relative localization techniques, the INS accumulates a drift with the time as the change in position is estimated by integrating the acceleration with time [166]. Depending on the required accuracy, IMU-based localization systems require an additional absolute sensing mechanism such as GNSS [167].

Laser systems application for position estimation has increased lately with the promising results of Light Detection and Ranging (LiDAR) sensor for this task [168]–[170]. The LiDAR is a relative distance measure technology that analyzes the light reflected when a laser is directed towards a target [171]. However, LiDAR-based localization systems are computationally costly and are sensitive to surface types and textures as the reflections of some surfaces may lead to unreliable data [172].

Furthermore, precise GPS, INS, or LiDAR-based localization systems cost can be higher than other sensors [173], [174]. Tschopp *et al.* [5] propose sensing infrastructure replacement with cheaper sensors to get a more cost-effective solution.

2.5 Conclusion

The research in the railway domain has recently increased due to the transformation of railway vehicles toward autonomous driving systems. However, it is a starting research field, with little specific work in train localization and no focus on the cameras as the primary sensors. The first work appeared in 2019 [5] with a visual-inertial odometry algorithm. Later works have followed the same approach presenting train localization systems based on sensor fusion [49]–[54]. However, the application of camera-based algorithms in the railway domain has not been explored yet.

The SOTA of CV-based autonomous robot and vehicle localization algorithms indicates how the research works have transformed from mainly geometric approaches in the early 2000s to the proliferation of learning-based algorithms due to the Machine Learning advances. The problems faced by early algorithms, such as the scale inconsistency, have been focused on by learning the depth and optical flow from image sequences. Furthermore, as most recent learning VO/vSLAM algorithms are based on learning scene depth and flow between consecutive frames, it has become a research direction itself. However, these algorithms have to deal with other problems such as the perceptually tough scenario characteristics or the adaptability to unknown environments.

The literature review of VO/vSLAM usage in visually degraded environments and scenarios evidences the need for further research in this direction. State-of-the-art algorithms have been tested in standard scenarios, but their performance is not measured in scenarios with different circumstances (e.g., poor lighting conditions, non-Lambertian surfaces, textureless areas or dynamic environments). Therefore, the analysis of VO/vSLAM algorithms from robotics and autonomous vehicles performance in these scenarios is a required research field.

For that task, ORB-SLAM2 [91] and DF-VO [133] algorithms have been selected from the SOTA. As stated before, the learning-based DF-VO algorithm outperforms most learning-based state-of-the-art algorithms, while ORB-SLAM2 is the most referenced geometric algorithm. Moreover, these algorithms represent two distinct types of VO algorithms (learning-based and geometric).

3 Dataset generation for VO/vSLAM in an urban underground railway scenario

This chapter introduces the dataset requirements of this research. First, an analysis of most referenced datasets in VO/vSLAM community is carried out. Next, as no standard or reference railway dataset fitted to the underground railway scenario was identified, a proprietary dataset generation method is explained.

3.1 Datasets for VO/vSLAM

The evaluation of any algorithm requires a labeled dataset from the application domain. Furthermore, having an adequately labeled ground truth in these datasets is especially essential for supervised Deep Learning approaches. This research focuses on the urban underground railway domain, and a dataset from the target domain is required. In this case, first, the most referenced datasets for VO/vSLAM were analyzed to find if any of them could fit the objective domain. In Table 3.1, the most referenced datasets in CV and localization research works are resumed. For each dataset, the domain it belongs to, the recording sensor configuration, how the 6-DoF pose has been obtained, and if the recording scenario is indoors or outdoors is extracted. This information is helpful to identify possible datasets when experimenting with VO/vSLAM algorithms.

From Table 3.1, it can be seen that most of the datasets belong to the robotics or car domains. Additionally, many datasets where the domain is the handheld sensor can be also found. Most of the datasets fuse several sensors when recording, being the stereo camera and the IMU the most used. Moreover, some of them include also a laser or RGB-D camera for depth measurement. The pose ground truth is mainly obtained from a GPS or a Motion Capture System (MoCap).

Most state-of-the-art VO approaches [71], [100], [116], [119], [124], [133] are evaluated in the standard KITTI vision benchmark proposed by Geiger *et. al.* [11],

Dataset	Domain	Sensor configuration	Pose ground truth	Environment
Cambridge Landmarks [61]	Handheld sensor	Monocular	SfM**	outdoors
7-scenes [175]	Handheld sensor	RGB-D	MoCap*	indoors
BigSFM [176]	Handheld sensor	Monocular	GPS	outdoors
ICL-NUIM [177]	Handheld sensor	RGB-D	SLAM	indoors
ADVIO [178]	Handheld sensor	Stereo/IMU	IMU	in/outdoors
OIVIO [179]	Handheld sensor	Stereo/IMU	Total station	in/outdoors
Rawseeds [180]	Robot	Stereo/IMU	GPS	in/outdoors
SUN3D [181]	Robot	RGB-D	SfM**	indoors
TUM-VI [182]	Robot	Stereo/IMU	MoCap*	in/outdoors
TUM-RGB-D SLAM [183]	Robot	RGB-D	MoCap*	indoors
TUM-Monocular VO [184]	Robot	Monocular	LSD-SLAM/MoCap*	in/outdoors
NavVis [103]	Robot	Monocular	GPS	indoors
MIT Stata [185]	Robot	Stereo/RGB-D/Laser	Laser	indoors
The Wean Hall [186]	Robot	Stereo/IMU/Laser/Wheel odometry	GPS	in/outdoors
RGB-D SLAM [187]	Robot	RGB-D	MoCap*	indoors
ETH3D [188]	Robot	Stereo/RGB-D/Laser/IMU	MoCap*/SfM**/LIDAR	in/outdoors
NCLT [189]	Segway	Stereo/IMU/Laser	GPS/IMU/Laser	in/outdoors
KITTI [11], [160]	Car	Stereo/IMU/Laser	GPS/IMU	outdoors
Málaga Urban [17]	Car	Stereo/IMU/Laser	GPS	outdoors
Oxford RobotCar [18]	Car	Stereo/Laser	GPS	outdoors
Ford Campus [19]	Car	Stereo/Laser/IMU	GPS	outdoors
KAIST Urban [20]	Car	Stereo/IMU	GPS/Laser	outdoors
Nordland [190]	**Railway**	**Monocular**	**GPS**	**outdoors**
Zurich Urban [191]	MAV	Monocular/IMU	GPS	outdoors
EuroC/MAV [12]	MAV	Stereo/IMU	MoCap*/Laser	indoors
MVSEC [192]	Multi Vehicle	Stereo/IMU/Laser	GPS/MoCap*/Laser	in/outdoors

*MoCap=Motion Capture System. **SfM=Structure From Motion

Table 3.1.: Referenced datasets for Computer Vision-based VO/vSLAM approaches ordered by domain or motion type.

[160]. This benchmark includes several datasets for tasks like VO, optical flow estimation, 3D object detection, or 3D tracking. The data is captured from a moving car in outdoor urban scenarios, and they provide datasets and evaluation metrics for each task. However, as the KITTI odometry dataset contains images from an outdoor environment with good lighting conditions, it is not adequate to evaluate the VO algorithms in the pursued scenario. Among the other analyzed datasets, it should be noted that only one database (Nordland [190]) covers the railway domain. However, Nordland covers outdoor railway scenarios, which is also out of the scope of this research work. Searching for a publicly available VO dataset from an indoor urban railway domain, no dataset was found. Therefore, the generation of a proprietary database was considered.

The data for a proprietary dataset can be collected from real and/or simulated sources. Real environment datasets are based on real-world scenarios, and therefore, the performance of algorithms can be effectively evaluated in the target scenario. However, the database generation in real-world scenarios increases recording and processing time, effort, and cost. In addition, it also depends on having access and permission to make the recordings in the target scenario (i.e. as in an underground railway substation).

Simulated environments can overcome these problems. The drawback of simulated environments is that it can not be assured that an algorithm trained and validated in a simulated environment will perform the same way in a real-world scenario. As stated in [193], all the challenging conditions inherent to underground environments can not be recreated in virtual scenarios.

Consequently, as a real-world underground railway scenario was accessible, a proprietary dataset was generated from a real underground railway scenario. The proprietary datasets' definition, generation and validation process are further explained in the following sections. First, the recording environment and setup, including camera calibration, are shown. Then, the ground truth generation method is defined.

3.2 Proprietary datasets: use case environments and recording setup

3.2.1 Use case environments

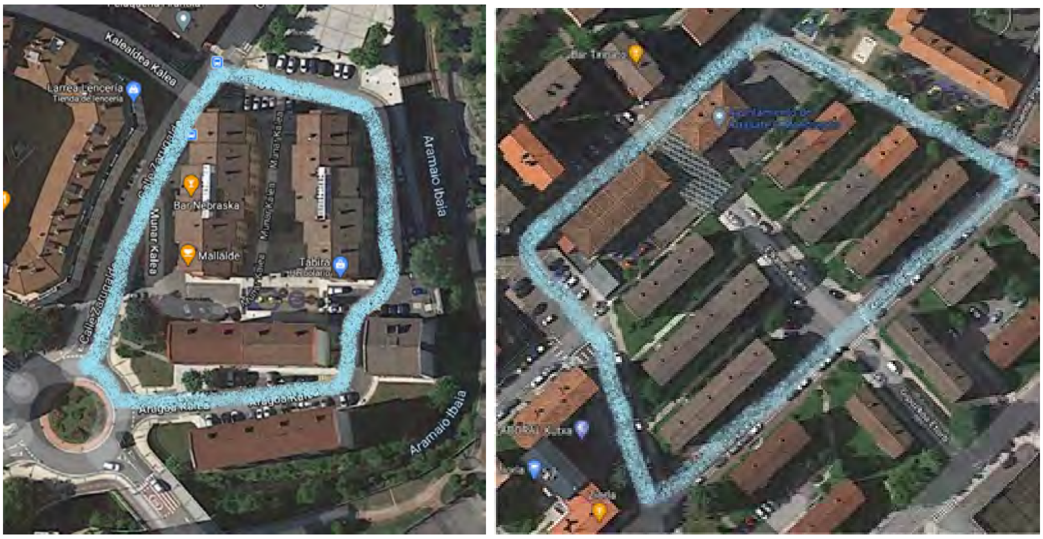
Before generating the proprietary dataset, a verification of the experimental setup was performed to verify the camera setup process. For that, a more accessible urban scenario was selected: a driving car scenario. Therefore, two use cases have been selected in this research. The first use case, *CarDriving*, has been used to verify the experimental setup. The second use case is the main use case, based on the railway domain. It is the *CAF* use case.

CarDriving use case

The *CarDriving* use case is based on the recording made in an urban car driving experience. For the dataset generated for this use case, two distinct trajectories were chosen to increase the robustness of the results and conclusions obtained from the experimentation. Furthermore, the trajectories were chosen following the idea of trying to replicate the conditions of the sequences from the KITTI dataset (e.g., the lighting, the slope of the path,...)

Furthermore, these trajectories have circular paths with the same starting and ending points. This was done aiming to improve the pose estimation in consecutive algorithms iterations. Some VO/vSLAM algorithms include a loop closing mechanism that substantially improves localization estimations when loops are found in the trajectories.

The trajectories shown in figure 3.1 have been named from the name of the street where they were recorded: Aragoa and Musakola. Two sequences were recorded in each trajectory, one by day and one by night. The daylight sequences are used to verify that the algorithms' performance in scenarios with comparable conditions to standard datasets is similar. The night sequences simulate the poor lighting conditions found in the urban underground railway domain.



(a) Aragoa

(b) Musakola

Figure 3.1.: The selected two trajectories in *CarDriving* dataset. Both trajectories belong to the same city and are circular paths in roads between high buildings. *Aragoa* trajectory has 300m in length and *Musakola* has 450m.

CAF use case

The *CAF* dataset was recorded in an underground railway line of Euskotren-Bilbao, named *Line 3* (L3). The entire trajectory map of the recordings made in the underground line L3 is depicted in Figure 3.2.

L3 has seven stations from with a total trajectory length of 5.8km. It contains poor lighting conditions in tunnel areas and significant light changes in platform areas. Furthermore, the tunnels are challenging for feature extraction algorithms as the textures from these scenarios are repetitive and the low light hinders the feature extraction. Figure 3.3 shows two examples of these scenarios to figure out the distinct conditions found in both situations: (a) tunnel and (b) platform.

In the recordings, the camera was placed in the front of the train, inside the driving cabin, because, due to safety restrictions, it can not be placed outdoors. Figure 3.4 shows the camera placement from the cabin's point of view.

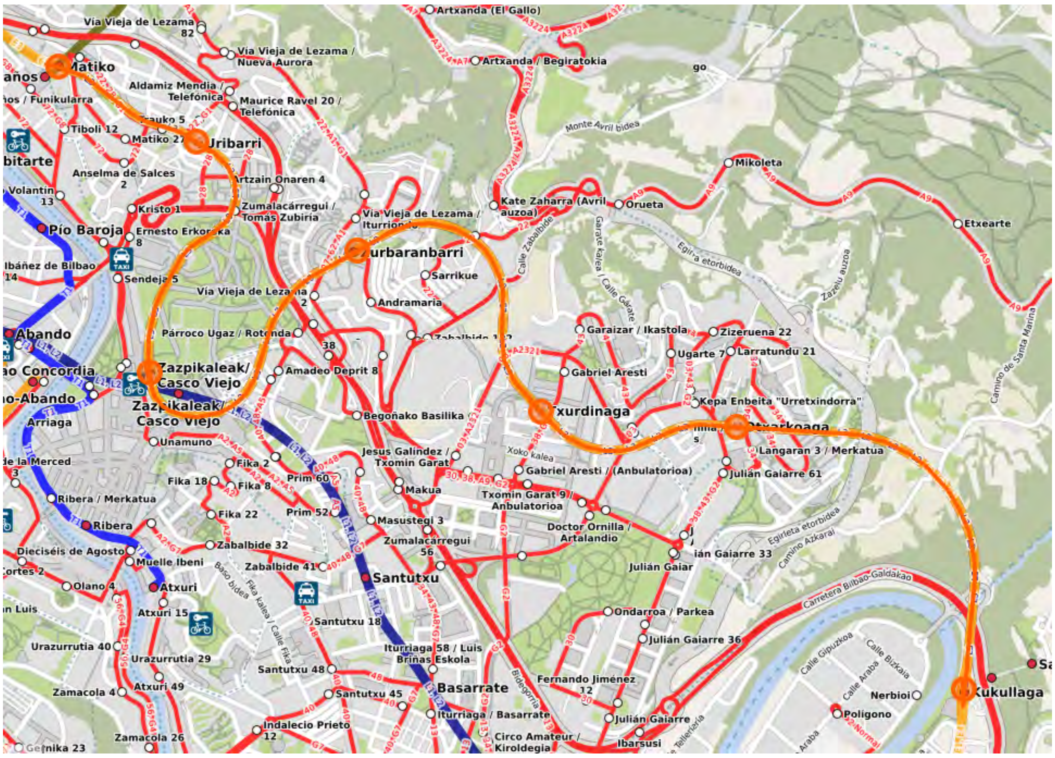


Figure 3.2.: Line 3 railway extracted from ÖPNVKarte map [194]. Each circle represents one station from L3. In total, L3 contains seven stations and trajectory length of 5.8km. The train circulates in both directions.

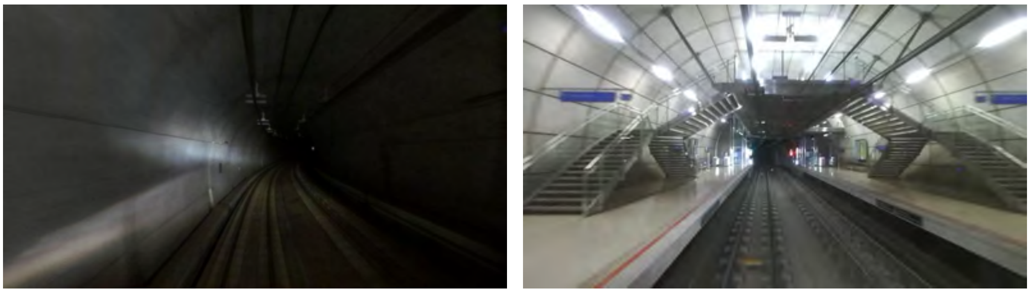


Figure 3.3.: The *CAF* dataset's tunnel and platform areas where the poor light conditions and textureless areas can be appreciated.

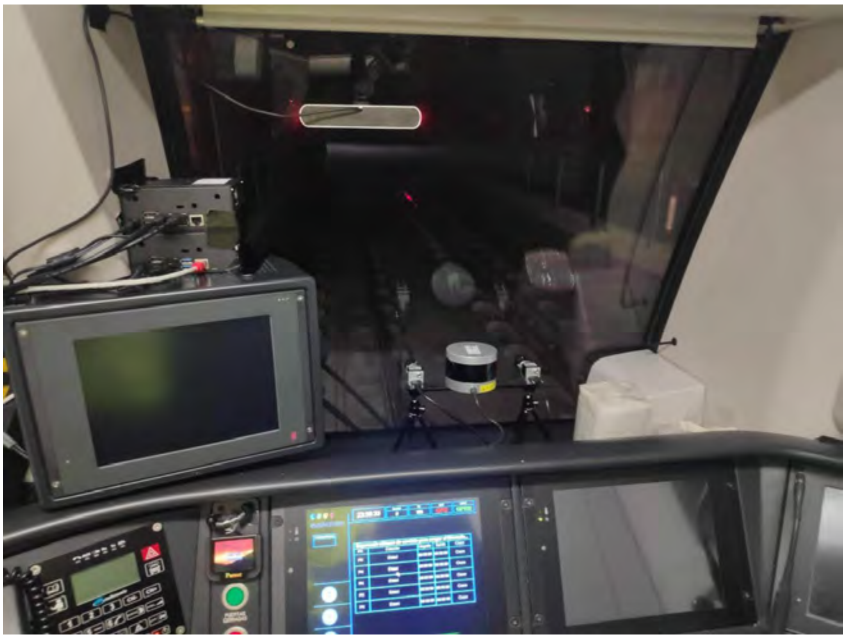


Figure 3.4.: Camera placement for *CAF* use case. Camera was placed in the front of a moving train in an urban underground railway scenario.

3.2.2 Recording setup

The base recording setup includes a camera and a laptop computer running the camera recording software (StereoLabs ZED API, ZED Explorer). The two hardware elements are connected by a high-rate USB3.0 wire. In the *CAF* use case environment, the ATP data monitored from the train is also included in the recording system. Figure 3.5 shows the diagram of the recording setup for the defined use case environments. As the base setup is comprised only by the camera and a laptop computer, it is adaptable to distinct use case environments and scenarios, and the introduction of other sensor is affordable.

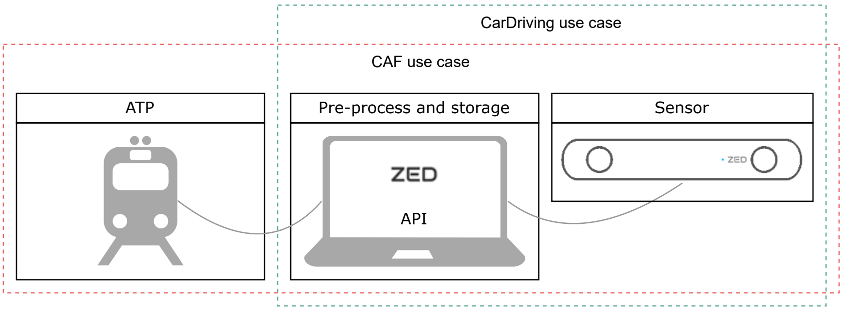


Figure 3.5.: Base recording setup comprised by the ZED stereo camera and a laptop computer. In the case of the *CAF* use case environment, train ATP data is also used.

The camera is a ZED Stereo Camera with an image resolution of 1280x720 pixels at 30 Hz, an electronic synchronized rolling shutter, automatic gain, and a lens aperture of F2.0. The computer is a Dell Latitude 5501, with Microsoft Windows 10 and the API provided by *StereoLabs* for the image pre-processing and storage. As shown in figure 3.6, the camera has been placed on the upper part of the front window of the vehicle (a car in the *CarDriving* use case and a train in the *CAF* use case), thus preventing the front of the vehicle from invading the lower part of the recording. In the case of the car, the camera has been also placed inside the vehicle aiming to replicate the same camera setup from the target railway scenario (in this scenario the camera must be placed inside the train due to safety restrictions).

Camera calibration is essential for VO/vSLAM algorithms since the images recorded through a camera contain a geometric distortion due to the used lens. In this case, the ZED camera API undistorts the images automatically when extracting them, and therefore, the images are undistorted directly and VO/vSLAM algorithms may be applied directly on them using the factory calibration parameters provided by *StereoLabs*. However, an independent calibration assessment was done to verify the camera calibration parameters. The calibration procedure is detailed in Section 5.2.

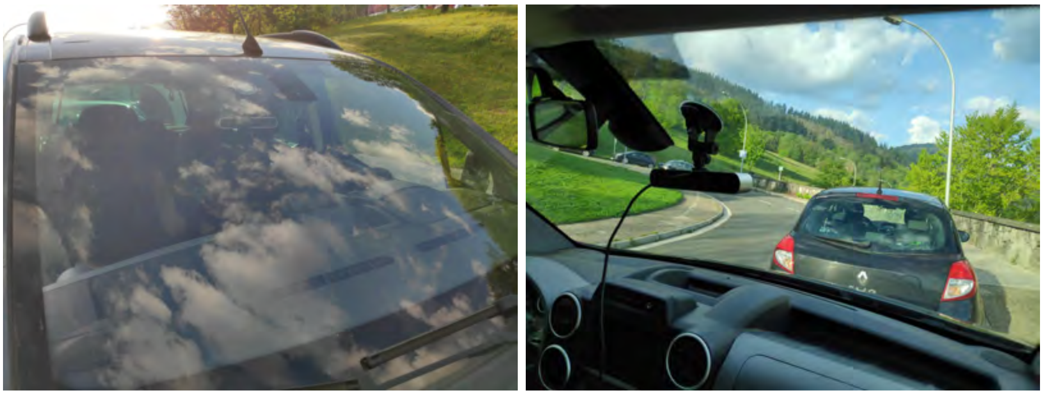


Figure 3.6.: Camera placements for *CarDriving* use case. The camera was placed in the top part of the windscreen of a car moving on an urban scenario.

In the following sections the recorded use case environments and each scenario's characteristics are explained.

3.3 Proprietary datasets: structure and ground truth

All the generated datasets are divided into sequences (seq). Each sequence contains a set of recorded images and a corresponding set of 6-DoF poses. The dataset follows the standard KITTI odometry dataset format and naming convention. The datasets structure adopts the following schema:

```
dataset
├── sequences/
│   ├── seq_n
│   │   ├── image_2/
│   │   │   ├── 000001.png
│   │   │   ├── 000002.png
│   │   │   └──     ⋮
│   │   ├── image_3/
│   │   │   ├── 000001.png
│   │   │   ├── 000002.png
│   │   │   └──     ⋮
│   │   ├── calib.txt
│   │   └── times.txt
│   ├── seq_n+1
│   └──     ⋮
└── poses/
    ├── seq_n.txt
    ├── seq_n+1.txt
    └──     ⋮
```

As the recordings were done with a stereo camera, the frames are stored in two folders, *image_2* for left camera frames and *image_3* for right camera frames. The frames are rectified RGB color images stored with loss-less compression using 8-bit PNG files. The size of the images is of 1280x720 (HD).

The camera calibration parameters are saved in a file named *calib.txt* file following the KITTI calibration format [11]. The calibration file contains the rectified projection matrix ($\mathbf{P}_{\text{rect}} \in \mathbb{R}^{3\times4}$) for each camera:

$$\mathbf{P}_{\text{rect}} = \begin{pmatrix} f_u & 0 & c_x & -f_u b_x \\ 0 & f_v & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

where (f_u, f_v) refers to the focal length, (c_x, c_y) is the principal point and bx is the baseline in meters w.r.t. reference camera 2. Given a stereo camera, the camera 2 and 3 represent the left and right cameras, respectively. The projection matrix is flattened for each camera i, where $i \in \{2, 3\}$, in the following way:

$$P\{i\} : f_u \; 0 \; c_x \; -f_u b_x \; 0 \; f_v \; c_y \; 0 \; 0 \; 0 \; 1 \; 0$$

This projection matrix represents a projection of a 3D point $\mathbf{p}_{\text{cam}} = (x, y, z, 1)^T$ from the rectified camera coordinates to an image point $\mathbf{p}_{\text{im}} = (u, v, 1)^T$.

The poses of a given sequence are stored in a pose file (*seq_n.txt*) that follows the KITTI convention: each row of the pose file contains the first 3 rows of a 4x4 homogeneous pose matrix flattened into one line. The homogeneous pose matrix p_n is represented as:

$$\mathbf{p}_n = [\text{r}_n | \text{tr}_n] = \begin{bmatrix} r11 & r12 & r13 & x_n \\ r21 & r22 & r23 & y_n \\ r31 & r32 & r33 & z_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where n is the frame number in the sequence, r_n and tr_n are the rotation matrix and the translation matrix of the n-th frame respectively. In the 6-DoF pose file it is represented flattened as:

$$r11 \; r12 \; r13 \; x_n \; r21 \; r22 \; r23 \; y_n \; r31 \; r32 \; r33 \; z_n$$

3.3.1 *CarDriving* dataset

As stated before, a proprietary VO dataset from a driving car has been generated to verify the experimental setup. Table 3.2 resumes this dataset. It is composed by two trajectories recorded in different street locations (refer to section 3.2.1). Each trajectory contains two sequences in two different lighting conditions (day, night). A sequence is the set of images recorded in the target trajectory. The quantity of images is depicted as *Frames* in Table 3.2. Finally, the distance traveled by the car in each recording is shown in the column *Sequence length*.

The reference ground truth data for *CarDriving* dataset has been estimated through the state-of-the-art geometric VO algorithm ORB-SLAM2 [195]. ORB-SLAM2 is

Trajectory	Sequence	Frames	Sequence length (m)	Lighting conditions
Aragoa	00	1150	300	Daylight
	01	1400		Night
Musakola	02	1640	450	Daylight
	03	1480		Night
Total	5670		750	

Table 3.2.: *CarDriving* dataset. The table contains the trajectory name, the sequence identifier, the frame quantity, the sequence length and the lighting conditions for each recorded sequence. The lihting conditions refer to the recordings being done by daylight or by night. Altogether, the dataset contains 4 sequences, 2 from each trajectory and with different characteristics.

widely used as a reference in the VO community [71], [196]–[198] and has been previously used as ground truth generation algorithm [106]. ORB-SLAM2 uses loop closure to relocalize the camera, and thus, improve the precision of the inferred path in successive iterations. Consequently, the previously mentioned recordings were done in closed paths where the starting and arrival points are the same.

As ORB-SLAM2 could have some scale issues, the scale is then corrected considering the trajectory length extracted from the reference *Google Maps* [199] trajectories. The length of the trajectory estimated by ORB-SLAM2 is scaled to match the trajectory length from reference Google Maps. Figure 3.7 shows reference ground truth data generated from this method for the trajectory named as Musakola.

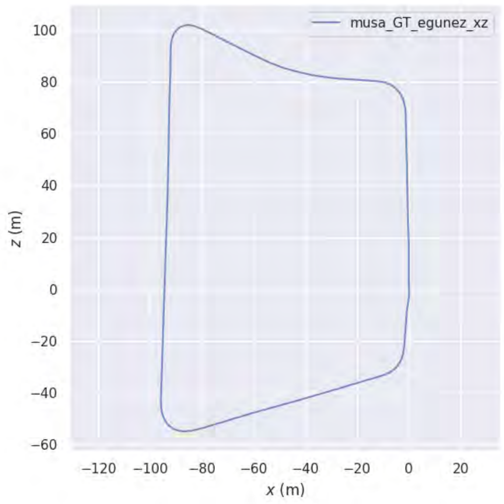


Figure 3.7.: Reference ground truth data generated using ORB-SLAM2 for Musakola trajectory.

3.3.2 *CAF* dataset

The *CAF* dataset comprises 19 sequences captured in the two directions of the rail Matiko-Kukullaga. A sequence is a trajectory between two stations defined as initial and arrival stations in Table 3.3. The dataset has been generated in two recording sessions of about five hours each. The dataset generation process is explained focusing on a sequence. A 6-DoF pose is estimated for each captured frame.

The generated dataset is represented in Table 3.3 where the sequences code, traveling direction, the initial and arrival stations for each sequence, and the number of frames for each sequence are depicted. The entire set of sequences yields 65.384 frames, with varying train speed and trajectory length. The quantity of frames of the dataset has been decided following the standard KITTI odometry dataset, defining a dataset that contains a similar volume of a standard VO benchmark dataset (e.g., KITTI).

Direction	Initial and arrival stations	Sequence	Frames	Length (m)
Matiko	Kukullaga-Otxakoaga	01_50	3048	1420.42
	Otxarkoaga-Txurdinaga	01_53	1977	695.41
	Txurdinaga-Zurbaranbarri	01_54	2663	1029.75
	Kukullaga-Zurbaranbarri	03_49	6700	314505
	Zurbaranbarri-Zazpikaleak	02_22	3260	1011.58
	Zazpikaleak-Uribarri	01_15	2724	902.93
		02_25	2639	902.36
	Zurbaranbarri-Uribarri	03_54	5904	1914.38
	Uribarri-Matiko	01_17	2532	500.45
		02_27	2505	499.81
Kukullaga	Matiko-Uribarri	01_31	2140	524
	Uribarri-Zazpikaleak	01_33	2830	898.34
	Zazpikaleak-Zurbaranbarri	01_35	2494	1004.42
	Matiko-Zurbaranbarri	03_36	6560	2435.27
	Zurbaranbarri-Txurdinaga	01_37	2550	1034.19
	Txurdinaga-Otxarkoaga	01_39	2126	694.62
	Zurbaranbarri-Otxarkoaga	03_41	4493	1728.01
	Otxarkoaga-Kukullaga	01_40	4095	1405.63
		03_44	4144	1406.01
TOTAL			65384	23152.56

Table 3.3.: *CAF* dataset resume with recorded sequences, the direction of the sequences, arriving station for each sequence, frame quantity, and sequence length. Overall, the dataset includes 19 sequences in the two directions, where the train travels between the different stations in the L3 line.

Most VO ground-truth datasets are generated using a GPS sensor [11], [17]–[20] (refer to Table 3.1). However, the GPS signal is not available in the underground zones. Thus, a method that computes the 6-DoF pose of each frame from the

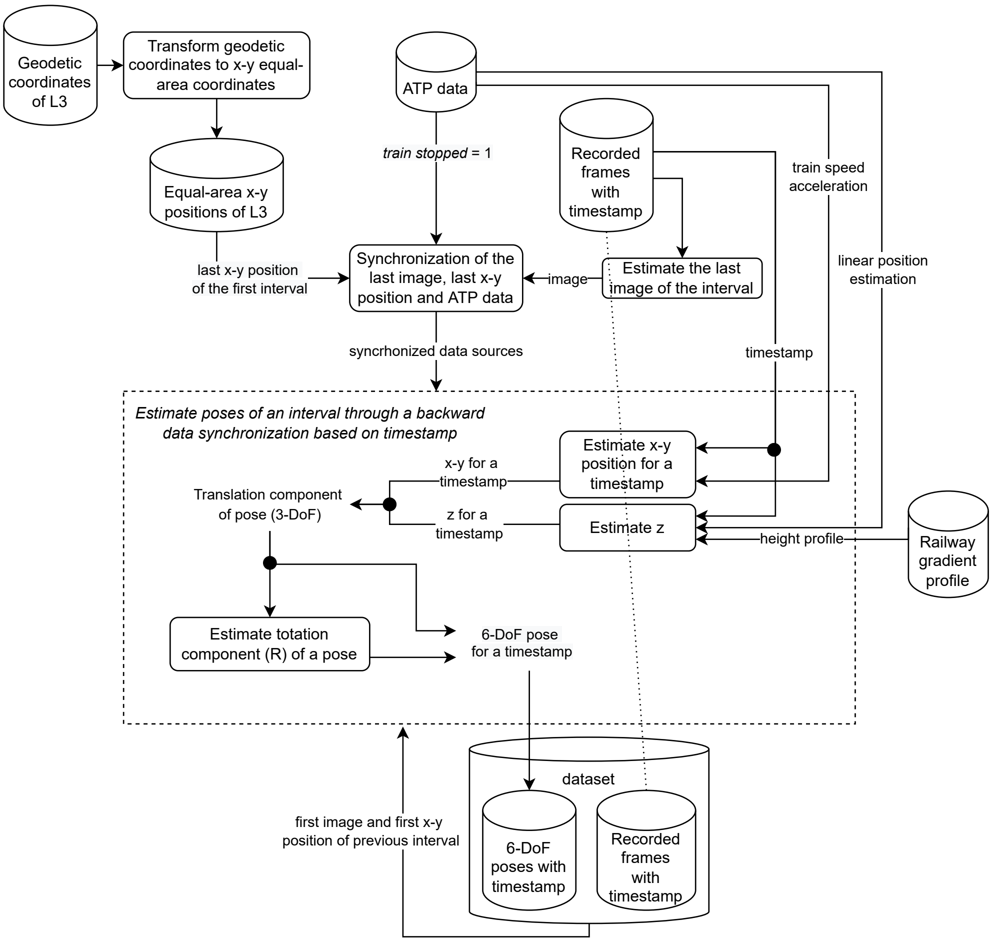

Geodetic coordinates of L3
Transform geodetic coordinates to x-y equal-area coordinates
ATP data
Equal-area x-y positions of L3
train stopped = 1
Recorded frames with timestamp
train speed acceleration
linear position estimation
last x-y position of the first interval
Synchronization of the last image, last x-y position and ATP data
image
Estimate the last image of the interval
syncrhonized data sources
timestamp
Estimate poses of an interval through a backward data synchronization based on timestamp
x-y for a timestamp
Estimate x-y position for a timestamp
Translation component of pose (3-DoF)
z for a timestamp
Estimate z
height profile
Railway gradient profile
Estimate totation component (R) of a pose
6-DoF pose for a timestamp
dataset
first image and first x-y position of previous interval
6-DoF poses with timestamp
Recorded frames with timestamp

Figure 3.8.: Schematic representation of the ground truth generation algorithm. The algorithm takes the geodetic coordinates of the railway, train ATP data and the railway gradient profile and outputs a ground truth pose for each recorded frame.

The algorithm first estimates (x,y) positions for a given sequence based on geodetic coordinates, then z is added through the gradient profile. Afterward, the (x,y,z) translation data is estimated for each frame using ERTMS ATP data, and, finally, the rotation data of each pose is calculated. In the following sections the data sources for the ground truth generation algorithm are explained.

Geodetic coordinates

The geodetic coordinates are represented by a pair (ϕ, λ) expressing *Latitude (Lat.)* and *Longitude (Lon.)* in decimal degrees. These coordinates use an ellipsoid to approximate the earth's surface locations [200]. The *Lat.* represents the angle between the normal and the equatorial plane while the *Lon.* measures the angle between the prime meridian (Greenwich meridian) and the measured point. The *Lat.* and *Lon.* are bounded by $\pm 90°$ and $\pm 180°$ respectively.

The geodetic coordinates define the coordinates of each position of the railway track and have been extracted from a Geomap called ÖPNVKarte [194]. This Geomap contains public data that includes worldwide public transport facilities (train, railway, ferry or bus). It is derived from OpenStreetMap [201], an initiative to create and provide accessible geographic data (i.e. street maps, etc.). It also contains railway-related information, such as platforms, stop positions, routes, and track positions.

The database is enriched with different institutions and external collaborators. The data for this proyect is extracted from the following institutions: Instituto Geográfico Nacional (IGN), Sistema Cartográfico Nacional (SCNE) and Eusko Jaurlaritza (Basque Government) through geoEuskadi institute and several external collaborators.

In the used Geomap, the minimum route track positions to define the L3 railway shape are known in geodetic coordinates (see figure 3.2). These track positions include the platforms start and ending (train stop location) positions, and positions between platforms. However, the camera frequency is higher than the geodetic coordinates discretization defined in the Geomap, and, therefore, a method based on monitoring and synchronizing the frame recording process, with train ERTMS ATP data has been designed and implemented to generate the poses of the frames that were recorded between the geodetic coordinates.

ÖPNVKarte is organized following its own tagging schema, *OpenRailwayMap* where information is divided into interrelated tags. Each tag is composed of several position nodes with *Lat.* and *Lon.* information. Table 3.4 shows the main tags from L3 line route where the reference coordinates are defined.

The geodetic coordinates of a sequence must be transformed from a 3D plane to a 2D plane to assign an equal-area (x,y) position to each geodetic coordinate. Figure 3.9 shows a sequence sample in geodetic coordinates and the generated equal-area

Tag	Description
Track	A track represents a railway in one direction and can contain multiple information with multiple sub-tracks. Each track is composed of nodes describing the trajectory in coordinate pairs.
Platform	Tag related to stations. Each station contains two platforms, one in each direction. It contains information such as name, position, area, if it is covered or if the platform is accessible with a wheelchair.
Stop	Nodes where the railway stops. Contains information related to the position and the railway type (light rail, subway, tram...).

Table 3.4.: Tagging scheme used by ÖPNVKarte map to classify the position data of rail transportation routes. The tags are used to define each rail positions and to generate hierarchical groups of geodetic coordinates.

(x,y) coordinates. In the ground truth generation algorithm, an equal-area (x,y) coordinate refers to x_n and y_n components of a 6-DoF pose.

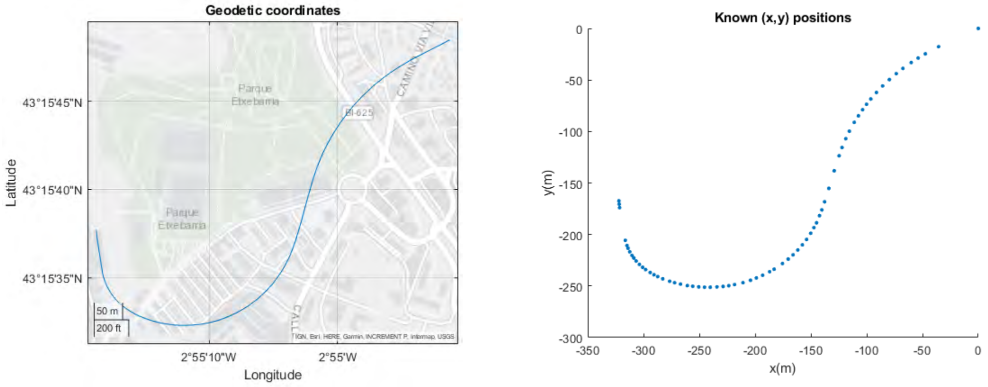


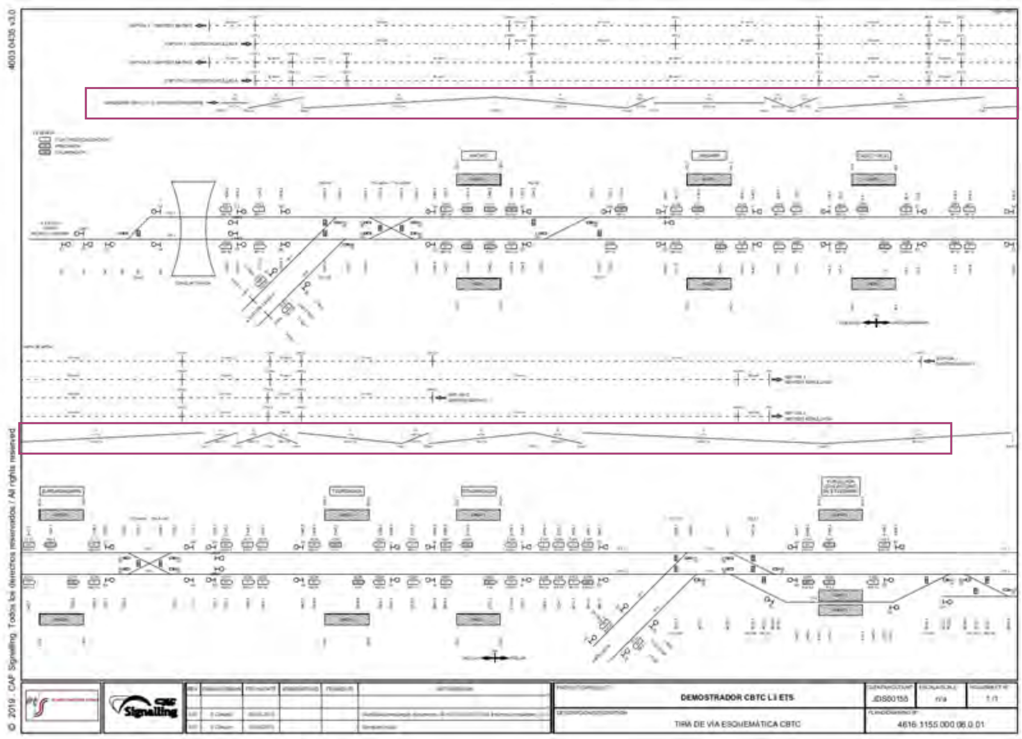
Figure 3.9.: Transformation of a L3 railway route positions defined by geodetic coordinates into equal-area (x,y) positions.

Railway gradient profile

The railway track gradient profile provided by the railway infrastructure managers defines how the railway track slope varies in predefined sections. These section lengths can be synchronized with ATP data to assign an slope for each meter of the railway. It allows the estimation of the height (z) of each 6-DoF pose. An height profile can be constructed with this gradient profile. The initial height is initialized as 0, and the height for each 1m section is calculated using the Equation 3.1.

$$h(d_n) = h(d_{n-1}) + (0.001 * grad_n) \tag{3.1}$$

where h refers to height, d_n refers to 1m trajectory sections, and $grad_n$ is the gradient value corresponding to that section from the gradient profile multiplied by 0.001 to adjust the units, as the gradient is taken in parts per thousands (‰). Figures 3.11 and 3.10 show the L3 railway gradient profile provided by the railway constructor and the estimated height profile, respectively. In the gradient profile provided by railway constructor, the gradient can be seen in the upper part of the railway definition ("*SSP VIAS 1 y 2 - SENTIDO CRECIENTE*") and it is defined in sections with known length and slope (being positive for increasing gradients and negative for decreasing gradients).



4003 0435 v3.0
© 2019 · CAF Signalling. Todos los derechos reservados / All rights reserved
DEMOSTRADOR CBTC L3 ETS
TIRA DE VIA ESQUEMATICA CBTC
4816 1155 000 00 0 01

Figure 3.10.: Gradient profile provided by the railway constructor. The gradients are marked in red, they can be seen in the upper part of the railway definition where the length of the different gradient sections is defined and the gradient is depicted. The positive gradients refer to sections with increasing slopes and vice versa.

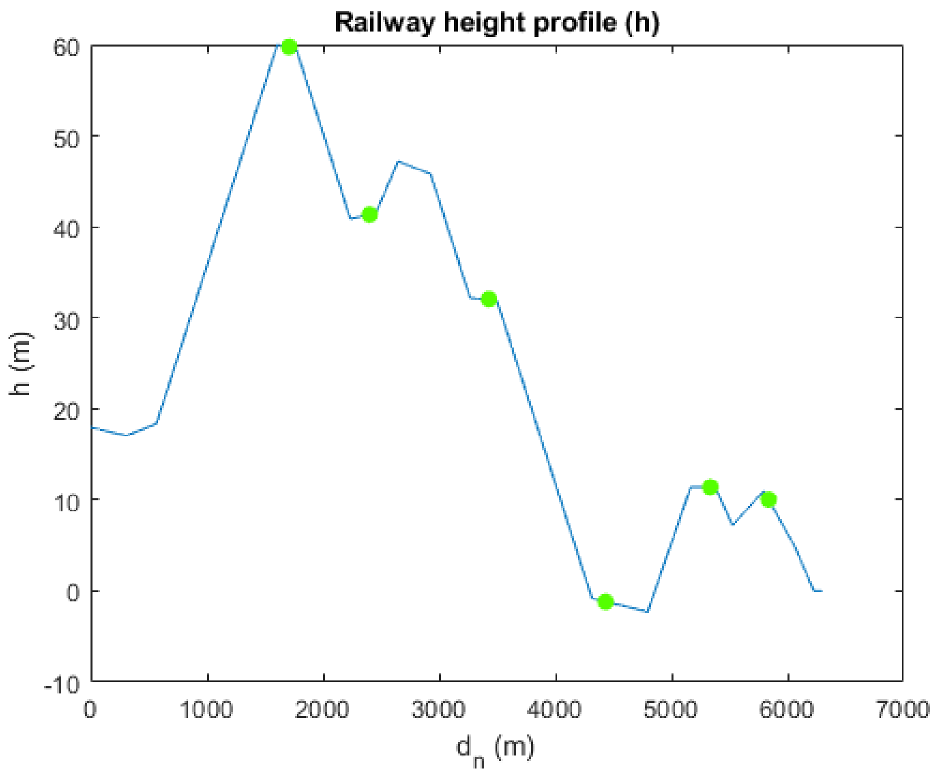


Figure 3.11.: Results of the height generation process. Height profile (h) is generated from gradient profile provided by railway constructor. The green circles represent the stations.

ATP data: train's dynamics and speed data

The ERTMS/ETCS ATP train speed estimation process is based on a redundant wheel encoder and radar sensors to get a safe and accurate estimation. Using these sensors, the ATP subsystem embedded in the train estimates the train position in the track, i.e. the distance traveled from a station or a beacon of the track. Track beacon position or inter-beacon distance is predefined and known by railway infrastructure managers, even by the ATP subsystem, and therefore the ATP train position is re-adjusted when a beacon signal is received obtaining a precise estimation. The 6-DoF pose estimation of each frame is made by synchronizing the ATP system monitoring process with the image recording process as both are installed in the train. This process aims to obtain synchronized ATP train position estimation for each frame. The data monitored from the ATP system is the following one:

- *timestamp* (s): time measured in the Coordinated Universal Time (UTC) standard read from the train's internal clock.

- *linear position estimation* (cm): distance traveled by the train from a previous station.
- *train speed* (m/s): train speed calculated by ATP.
- *train acceleration* (cm/s^2): train acceleration calculated by ATP.
- *train stopped*: boolean reflecting whether the train has reached stopping point or not.

All those variables are extracted from an ATP monitoring proprietary application that monitors train odometry data. The data acquisition frequency is higher than the camera frequency (30Hz), and, consequently, they have been synchronized and a pose estimated for each frame.

Estimate poses of an interval through a backward data synchronization based on the timestamp

The synchronization algorithm's main idea is to estimate poses between two geodetic coordinates (x,y). Those inner poses need to be computed in a way that can be synchronized with the camera's frame-rate. To do so, the section between two consecutive coordinates is defined as a straight line *interval*. Figure 3.12 represents a conceptualization of a given sequence with the intervals, the initial and arrival stations, the (x,y) positions and the estimated poses. The main concepts of the ground truth generation algorithm are described in Algorithm 1.

As the data is synchronized at the sequence ending, the synchronization process of an *interval* is done following backward trajectory and leveraging the timestamps.

Given a sequence, the last (x,y) position, the last frame and the ATP data are taken for a given interval and the poses for all frame timestamps in that interval are estimated. Then, the poses of the following interval are estimated by taking the last (x,y) position and the last frame of the previous interval as the initial position.

However, the train speed is variable and, therefore, the distribution of these poses can not be linear in different intervals. Therefore, the train speed is used to calculate the quantity of poses in each interval. The total number of poses within the whole sequence should match the recorded frame amount.

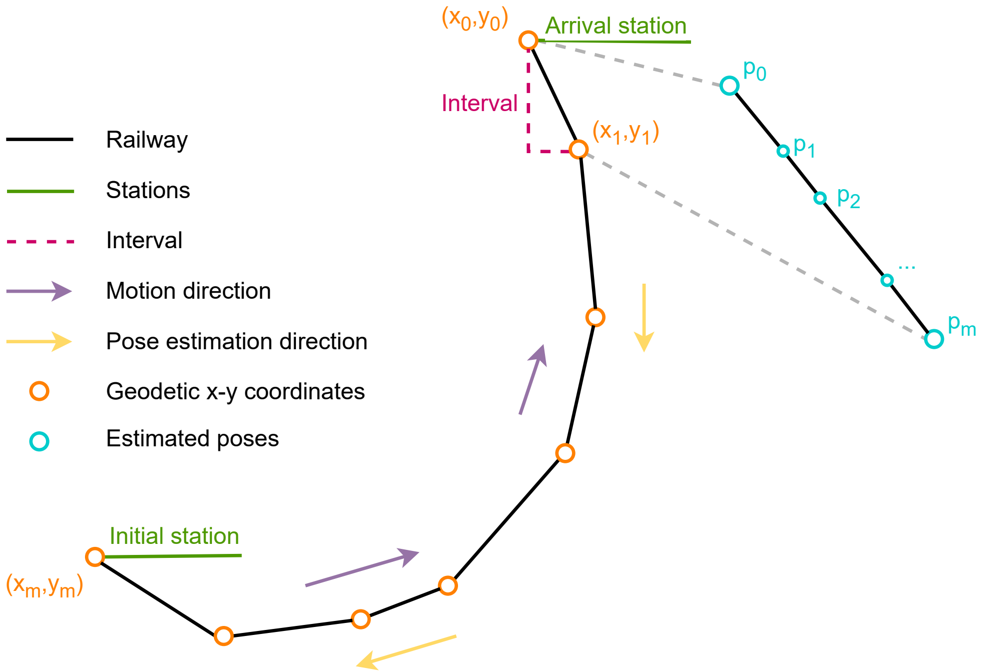


Figure 3.12.: A conceptualization of a example sequence with the (x,y) positions extracted from geodetic coordinates, the intervals, the initial and arrival stations and estimated poses.

Synchronize the last (x,y) position, last frame and ATP data. The first step is to synchronize the different data sources of a sequence using the last (x,y) position of the arrival station, last frame and ATP data. The algorithm generates ground-truth poses for each recorded sequence using the position where the train has started the trajectory at the initial station as origin. For that, first the image where train stops (last frame of the sequence) must be estimated. When there is motion, the similarity between consecutive frames is very low, however the similarity increases when the train has stopped. Due to the similarity of the frames corresponding to the train stopping point, the last frame is selected using the Structural Similarity Index (SSIM) [202].

SSIM is one of the most standard algorithms for image quality assessment [203], and therefore, for image similarity measure. It has shown that can outperform other common image similarity measurements as MSE [204] and has been intensely researched [205]. The SSIM measures the luminance, contrast, and structure of two given images and returns a similarity value between them.

Also, it is easily implementable and requires just a starting optimization phase where the threshold is selected. Furthermore, the index was used to find just the first image within the threshold in each sequence, which gives a little number of results totally.

Algorithm 1 Ground truth data generation algorithm

Input: Given an *interval* (i) defined as a straight line between two (x,y) positions

```
Phase 1 - Synchronize last (x,y) position, last frame and ATP data of an interval
if $i = 0$ then                                                          ▷ First interval
    Last frame ← $SSIM > threshold$                                      ▷ SSIM [202]
    Last $(x_i, y_i)$ position ← given in the interval definition
    ATP data ← $train\_stopped = 1$
else                                                                     ▷ Following intervals
    Last frame, last $(x_i, y_i)$ position and ATP data ← taken from $i - 1$
end if
Phase 2 - Estimate poses on an interval through a backward data synchronization based on the timestamps
Input: $V_n$: train speed, $a_n$: train acceleration, $t$: timestamp, $h$: height profile, $d_n$:
    linear position estimation
Estimate translation component of poses ($\mathrm{tr}_n$)
    a: $(x_n, y_n) \leftarrow f(v_n, a_n, t)$                            ▷ Eqn. 3.3
    b: $z_n \leftarrow h(d_n)$                                           ▷ Eqn. 3.1
Estimate rotation component of poses ($\mathrm{r}_n$)
    a: $\mathrm{r}_n \leftarrow g(\mathrm{tr}_{n-1}, \mathrm{tr}_n)$     ▷ Eqn. 3.4, 3.5, 3.6
```

Although SSIM is sensitive to image distortions, the environment being static, and the view fixed enables the SSIM application in underground railway scenarios. The SSIM between two images I_n and I_{n+1} is calculated following the Equation 3.2. This equation measures the luminance, contrast and structure of both images.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)((2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3.2}$$

where $c_1 = 2.56$ and $c_2 = 7.68$. The threshold was selected by testing. A threshold was proposed and then updated until a SSIM threshold that best fitted to the lighting conditions of the scenario was found. It was a handmade process where the results were evaluated by this paper authors. In this case a $SSIM > 0.965$ has been used as similarity threshold at the train stopping point. The first frame from a given sequence with a SSIM value higher than the threshold was selected as the train stopping frame.

The last (x,y) coordinates refer to the train stopping position at the arrival station; therefore, this coordinate pair and the last frame are already synchronized. Finally, ATP monitored data is synchronized using the *train stopped* variable.

Estimate poses of an interval through a backward data synchronization based on timestamps. This process has two steps; first, the translation component is estimated, and then, the rotation is calculated from that translation.

Estimation of translation component. Translation component $\mathbf{T} = \{\mathrm{tr}_0, \mathrm{tr}_1, ..., \mathrm{tr}_m\}$ is defined as 3-DoF poses ($\mathrm{tr}_n = [x_n, y_n, z_n]$) of an interval where n is the pose number ($0 \leq n \leq m$) and m is the total number of poses for that interval. For the translation component of a pose, first, the (x,y) position is estimated, and then the height (z) is added. The translation is estimated by taking an initial (x,y) position and calculating the motion to the next (x,y) position using the ATP *train speed* and *train acceleration* data. The ATP data is not constant, and therefore, the frame timestamps and ATP data timestamps are synchronized to acquire the train speed and acceleration values corresponding to each recorded frame.

The translation between two consecutive (x,y) positions in a straight line that forms the interval can be calculated using *Uniformly Accelerated Motion (UAM)* equations. This estimation is possible because it is considered that the poses follow a motion in a straight line and with a constant acceleration between them. Equation 3.3 shows the application of UAM equations in this case.

$$d_n = v_{n-1}t + \frac{1}{2}a_{n-1}t^2 \tag{3.3}$$

where t refers to the timestamp, v_n and a_n refer to ATP data train speed and acceleration respectively. The initial (x,y) translation component is set as $[0, 0]$.

After calculating the (x,y) positions, the z or height is estimated using the height profile estimated from the gradient profile and the traveled distance from the ATP data. The height profile is based on fixed distances from an origin (see Figure 3.10. The railway height profile can be synchronized with the train stopping point, and therefore, with the first (x,y) position as the origin. Then, previously calculated (x,y) positions can be used to extract the Euclidean traveled distance from that origin to the following positions.

Each pose's height (z) is calculated using the traveled distance from the previous pose and the height profile. Therefore, after height estimation, the translation component (x,y,z) of a pose has been estimated with respect to a timestamp.

Estimation of rotation component. Rotation component $\mathbf{R} = \{\mathrm{r}_0, \mathrm{r}_1, ..., \mathrm{r}_m\}$ is defined as the rotation matrices (r_n) within an interval where n is the pose number ($0 \leq n \leq m$) and m is the total number of poses for that interval calculated in the previous steps.

To calculate the rotation component r_n for each translation tr_n the transformation between two consecutive orientation vectors or_{n-1} and or_n is estimated. or_n defines the orientation of the train in tr_n and represents the vector between consecutive translations tr_{n-1} and tr_n. It is calculated as shown in 3.4:

$$or_n(tr_{n-1}, tr_n) = (x_n - x_{n-1}, y_n - y_{n-1}, z_n - x_{z-1}) \tag{3.4}$$

where x, y and z represent the translation components of tr_{n-1} and tr_n. Then, using the axis-angle representation, the transformation between consecutive orientation vectors or_{n-1} and or_n can be calculated. For that, first the orientation vectors are normalized by dividing their value with the Euclidean norm (vector magnitude) $\|or_n\|$ of each vector (Eqn. 3.5) to align them at the same origin. The Euclidean norm can also be defined as the Euclidean distance of a vector from the origin to a point.

$$\hat{or}_n = \frac{or_n}{\|or_n\|} \tag{3.5}$$

Then, the Euclidean norm of the cross product between the normalized consecutive orientations is estimated to get the axis. Finally, the rotation component is estimated using the inverse cosine function as shown in equation 3.6, where the angle between the orientations vectors is calculated trough the dot product:

$$r_n = acos(\frac{\|\hat{or}_n \times \hat{or}_{n-1}\|}{\hat{or}_n \cdot \hat{or}_{n-1}}) \tag{3.6}$$

where $acos$ refers to the inverse cosine function and $\hat{or}_{n-1}$ and $\hat{or}_n$ to two consecutive orientation vectors. This rotation estimation method accumulates an error relative to the previous estimations. However, as the train is tied to the rails, the trains' orientation is always fixed, and the orientation estimation is not critical.

The previously calculated translation component is added to the newly calculated rotation component to obtain the target 6-DoF ground truth pose.

Once all the poses from a given interval have been estimated, the next interval is taken and the process is repeated until all the intervals of a sequence have been covered.

3.4 Conclusions

This chapter resumes the solution given to the data requirements of the research. The need for a proprietary dataset from the target domain arose when analyzing the most referenced datasets in state-of-the-art VO/vSLAM. The main dataset *CAF* was generated recording in an urban underground railway scenario.

Additionally, given the difficulties to access the target scenario, a supplementary dataset *CarDriving* has been recorded to validate the recording setup (e.g., camera setup, frame quality,...). This dataset was generated from a driving car in an urban environment trying to imitate the same recording conditions even recording at night to replicate the low-light conditions of the urban underground railway scenario.

The ground truth for the *CarDriving* dataset has been obtained from the standard algorithm ORB-SLAM2 and corrected with GPS data. In the case of the *CAF* dataset, ground truth has been generated by designing and implementing a new method to estimate 6-DoF poses for the frames recorded in an underground railway line. The generation process is based on the synchronization of geodetic coordinates, ATP data, and a railway gradient map.

To our knowledge, the *CAF* dataset is the first VO/vSLAM dataset in the urban underground railway domain. In order to make it scientifically sound, it follows the standard KITTI format and data volume. This dataset enables the testing of the state-of-the-art VO/vSLAM algorithms in scenarios with perceptual characteristics that can not be found in the standard VO/vSLAM datasets.

The next chapter addresses the VO/vSLAM performance in the target domain utilizing generated the proprietary datasets. The experimental setup for that analysis is also included.

4 VO/vSLAM in an urban underground railway scenario

This chapter evaluates the experimentation of the state-of-the-art VO/vSLAM algorithms' performance in an urban underground railway scenario is evaluated. First, the standard VO/vSLAM evaluation metrics are studied. Then, the experimentation setup is explained, and finally, the VO/vSLAM results in the proprietary datasets are discussed.

4.1 VO/vSLAM evaluation metrics

Traditionally, the performance of VO/vSLAM algorithms has been evaluated focusing on different aspects. In the case of feature-based VO/vSLAM algorithms where the standard pipeline is built around the feature detection and matching processes, some research works have concentrated on evaluating those processes[206]. For that, they have measured the extracted features quantity or quality, and the feature-matching performance. Other works have targeted the metrics to measure the estimated pose accuracy or the algorithm robustness by measuring the reprojection error, RANSAC iterations and the percentage of inliers [186], respectively. Most works evaluate VO/vSLAM pose estimation based on estimating the Mean Square Error (MSE) between the predicted and real poses, and, the Normalized Root MSE (NRMSE) and the Standard SME (SMSE) to analyze the estimation error of the results [95].

Based on those previous metrics, Geiger *et. al.* [160] introduced some metrics that have become standard metrics in the VO research community for the pose estimation evaluation. This thesis experimentation follows some of those metrics used by all the works proposed later. The metrics include comparisons of absolute pose measures through the Absolute Trajectory Error (ATE) and comparisons of relative pose measures using Relative Pose Error (RPE) [187].

A 6-DoF alignment is recommended to evaluate shape similarities of trajectories [207]. Therefore, all the trajectories have been transformed with a 6-DoF Umeyama alignment [208], a standard alignment method followed by most publications in *KITTI Visual Odometry / SLAM evaluation benchmark* [160].

Given this transformation, ATE evaluates the global consistency of estimated translations compared to the ground-truth trajectory. The ATE is measured in meters. The ATE for a given sequence is obtained by calculating the root-mean-square error (*RMSE*) over all timestep i with respect to the reference data as in Equation 4.1:

$$ATE_{trans} = \left(\frac{1}{n} \sum_{i=1}^{n} \left\| Q_i^{-1} P_i \right\|^2 \right)^{\frac{1}{2}} \tag{4.1}$$

where P_i and Q_i are the estimated pose, and ground truth pose at the time i, respectively, and n is the total number of poses of the sequence.

For relative VO evaluation, the RPE measures the drift error for each pose of the trajectory. The rotation and translation components are calculated separately in the RPE calculation. It is measured in meters for the translation component. For rotation, the rotation matrix is converted to angle-axis representation and the rotation angle is used as the error. Then, the RMSE for each component is calculated using Equation 4.2:

$$RPE = \left(\frac{1}{m} \sum_{i=1}^{m} \left\| \left(Q_i^{-1} Q_{i+1} \right)^{-1} \left(P_i^{-1} P_{i+1} \right) \right\|^2 \right)^{\frac{1}{2}} \tag{4.2}$$

where P_i and Q_i are the estimated pose, and ground truth pose at the time i, respectively, and m is the total number of relative poses from a sequence with n camera poses, where $m = n - 1$.

Finally, following the VO evaluation criteria of the KITTI evaluation benchmark, the Average Translational Error (t_{err}) and the Average Rotational Error (r_{err}) are calculated on sub-sequences of different lengths. These errors measure the average relative pose error at a fixed distance. The sub-sequences length in meters is (100,200,...,800) because the error for smaller sub-sequences affect significantly and do bias the evaluation results. The t_{err} is measured in percent and the r_{err} in degrees per meter.

4.2 Experimentation setup

DF-VO implementation was taken from [209], and weights of flow estimation deep models were selected from the authors' trained models. The flow model is initially trained in the synthetic dataset Scene Flow [210] with good generalization ability. The model *stereo_640x192*, previously trained in the ImageNet and KITTI datasets, is used as the depth estimation model for Monodepth2 [157].

In the case of ORB-SLAM2, in order to handle its non-deterministic nature, each sequence is executed five times. The median accuracy of the estimated trajectory is evaluated as proposed by authors in [91]. Each ORB-SLAM2 execution is different because the initialization process is based on the reliability of descriptors matching.

The VO evaluation is done using the *KITTI Odometry Evaluation Toolbox* as in [133], in the same workstation used for training.

4.3 DF-VO in *CarDriving* dataset

To verify the recording setup, DF-VO [133] has been evaluated in *CarDriving* sequences. The results have been compared to the results obtained by the same algorithm in the standard KITTI dataset (see Table 4.2).

Trajectory	Aragoa		Musakola		Avg. Err.
Seq.	00	01	02	03	
t_{err} (%)	7.81	8.26	9.70	12.52	9.572
r_{err} (°/100m)	3.22	4.36	3.18	5.92	4.17
ATE (m)	5.21	5.71	3.83	8.73	5.285
RPE (m)	0.269	0.268	0.305	0.38	0.305
RPE (°)	0.785	0.784	0.398	0.565	0.633

Table 4.1.: Results obtained from DF-VO application in the *CarDriving* dataset following the standard metrics.

As shown in Table 4.1, the results in the daylight sequences (00 and 02) are better than sequences recorded by night (01 and 03). The minimum ATE is obtained in the sequence 02 (3.86m). The scenario where sequence 02 was recorded has similar characteristics (e.g., more building and static objects) to the KITTI dataset. On the contrary, sequences 00 and 01 have some open spaces with scenarios with less features (e.g., the sky) and dynamic objects that can be challenging for VO/vSLAM methods.

The average translational (t_{err}) and rotational (r_{err}) errors are higher than the errors obtained in the KITTI dataset, as the sequences lengths are shorter and, therefore, the sub-sequences used to calculate these metrics have been shorter $(10, 20, 30, 40, 50, 60, 79m)$. As the authors of the proposed metrics explain in [160], the average errors increase in short sub-sequences.

The results shows that when the scenario's light conditions are similar to the lighting of standard datasets, the results are also similar. The poor light conditions in the sequences recorded by night hinder the DF-VO performance.

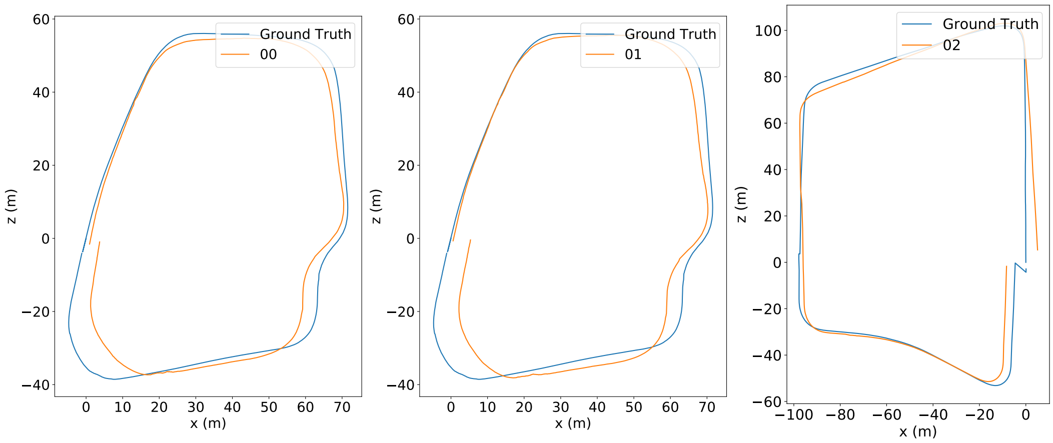
Algorithm	Metric	Avg. Err
DF-VO in KITTI dataset	t_{err} (%)	1.972
	r_{err} (°/100m)	0.365
	ATE (m)	6.872
	RPE (m)	0.041
	RPE (°)	0.038

Table 4.2.: Average standard VO errors obtained by DF-VO in the standard KITTI dataset [133].

From Table 4.1, it can be seen that the average ATE obtained over all the sequences of 300/450m long is 5.285m. Consequently, the results obtained by DF-VO in the *CarDriving* dataset are similar to the results obtained by DF-VO in the KITTI dataset in the sequences where lighting conditions and scenario characteristics (e.g., the appearance of buildings, no big slopes and static objects) are also similar.Therefore, the recording setup is considered validated for the later experimentation.

However, the night recorded sequences trying to replicate poor lighting conditions found in the underground railway scenario obtain slightly worse results and, therefore, DF-VO is expected to follow the same behavior in the *CAF* dataset. Figure 4.1 shows the trajectory results of DF-VO against the ground truth trajectory generated by the ORB-SLAM2 (see section 3.3.1 for the ORB-SLAM2 based ground truth generation). Sequence 00 (daylight) and 01 (night) were recorded in Aragoa and sequence 02 was recorded in Musakola by day.

From Figure 4.1, it can be seen that the trajectories estimated by DF-VO are close to the reference trajectories.



(a) Sequence 00 (day) from Aragoa.

(b) Sequence 01 (night) from Aragoa.

(c) Sequence 02 (day) from Musakola.

Figure 4.1.: Comparison between DF-VO pose estimation and the reference created by ORB-SLAM2 in *CarDriving* dataset.

4.4 DF-VO and ORB-SLAM2 in *CAF* dataset

This section discuss the evaluation of the state-of-the-art VO/vSLAM algorithms' performance in the *CAF* proprietary dataset. The ground truth of this proprietary dataset has been generated by synchronizing geodetic coordinates, train ERTMS ATP data and the gradient profile of the target railway (see section 3.3.2). Table 4.3 shows the errors for each sequence estimated with selected VO algorithms DF-VO and ORB-SLAM2. Figures 4.2 and 4.3 represent the errors depicted in Table 4.3.

Previously, DF-VO and ORB-SLAM2 were evaluated in the KITTI Odometry dataset; however, KITTI does not contain those perception challenges as it contains considerably different properties related to the sequence length and visual characteristics.

In Table 4.3, it can be seen that ORB-SLAM2 obtains better results than DF-VO in all the metrics. The minimum ATE is found in sequence *01_17* (15.61m), the shortest sequence, with a length of 505m. On the contrary, the maximum ATE is estimated by DF-VO in the sequence *03_41* (478.76m) from a 1729m length sequence. The smallest translational RPE is estimated by ORB-SLAM2 (0.086m in the *01_17* sequence), while the largest is estimated by DF-VO (0.764m in the sequence *01_37*). The sequence length affect to the error estimations in this scenario.

The average translational (t_{err}) and rotational (r_{err}) errors follow the same behavior. An average t_{err} of 70.07% and 58.34% are obtained by DF-VO and ORB-SLAM2, respectively. These results are higher than those obtained by the same algorithms in the standard KITTI dataset (see Table 4.4. However, the urban underground

Algorithm	Record	14_11_2021 (->Matiko)									
	Seq	01_50	01_53	01_54	03_49	02_22	01_15	02_25	03_54	01_17	02_27
DF-VO	t_{err} (%)	70.41	51.59	64.62	85.76	52.4	54.58	60.02	88.82	57.01	53.5
	r_{err} (°/100m)	10.28	15.42	19.85	13.11	18.78	16.23	15.75	20.64	17.12	18.2
	ATE (m)	230.66	106.38	157.46	478.76	135.36	29.11	94.27	175.64	26.1	36.39
	RPE (m)	0.407	0.23	0.329	0.43	0.254	0.244	0.318	0.379	0.095	0.096
	RPE (°)	0.157	0.135	0.156	0.117	0.124	0.13	0.157	0.143	0.057	0.062
ORB-SLAM2	t_{err} (%)	42.95	35.21	43.17	86.18	34.55	43.15	46.47	89.62	32.91	31.92
	r_{err} (°/100m)	7.72	12.88	15	10.5	13.34	15.16	16.24	17	8.3	8.22
	ATE (m)	56.58	44.98	40.54	355.95	72.35	26.71	74.61	177.23	15.61	17.1
	RPE (m)	0.351	0.154	0.277	0.597	0.256	0.215	0.307	0.416	0.086	0.085
	RPE (°)	0.081	0.092	0.102	0.106	0.125	0.104	0.11	0.121	0.065	0.064

Algorithm	Record	14_11_2021 (->Kukullga)									Avg. Err.
	Seq	01_31	01_33	01_35	03_36	01_37	01_39	03_41	01_40	03_44	
DF-VO	t_{err} (%)	38.32	79.46	130.71	70.63	72.48	61.13	85.75	81.28	72.91	70.0726
	r_{err} (°/100m)	10.71	23.49	35.54	16.12	23.52	15.55	14.32	26.17	29.29	18.9521
	ATE (m)	38.38	104.98	111.69	343.44	150.64	62.88	270.45	226.86	114.75	152.3263
	RPE (m)	0.149	0.374	0.764	0.385	0.438	0.257	0.408	0.322	0.34	0.3273
	RPE (°)	0.088	0.13	0.584	0.116	0.166	0.103	0.111	0.103	0.147	0.1497
ORB-SLAM2	t_{err} (%)	32.66	78.95	119.83	71.01	66.8	47.76	94.18	55.47	55.8	**58.3468**
	r_{err} (°/100m)	7.38	16.5	17.34	14.55	16.44	13.31	11.09	10.57	10.77	**12.7531**
	ATE (m)	30.67	38.47	88.96	172.66	36.31	22.28	231.15	103.74	98.84	**89.7231**
	RPE (m)	0.127	0.388	0.613	0.381	0.463	0.255	0.414	0.3	0.3	**0.3171**
	RPE (°)	0.09	0.099	0.113	0.11	0.113	0.079	0.097	0.07	0.07	**0.0953**

Table 4.3.: DF-VO and ORB-SLAM2 application evaluation using standard VO evaluation metrics: average translational (t_{err}) and rotational (r_{err}) errors, ATE and RPE. The sequences are organized by the direction they are recorded. The average errors for all 19 sequences are calculated, and the best result is in bold.

railway scenario characteristics are distinct and hinder the use of light-dependent localization algorithms, such as DF-VO and ORB-SLAM2.

In some of the sequences, much larger ATE than in the other sequences can be appreciated (e.g., sequences 03_49, 03_36, and 03_41). The length of those sequences is larger than others (see dataset definition in Table 3.3) because they skip some stations (e.g., sequence 03_49 skips stations *Otxarkoaga* and *Txurdinaga*, and stops at *Zurbaranbarri*). As all the used metrics are incremental and the sequences do not contain loops (do not go through the same place two times) for global localization optimization, the trajectory length seems critical for pose estimation.

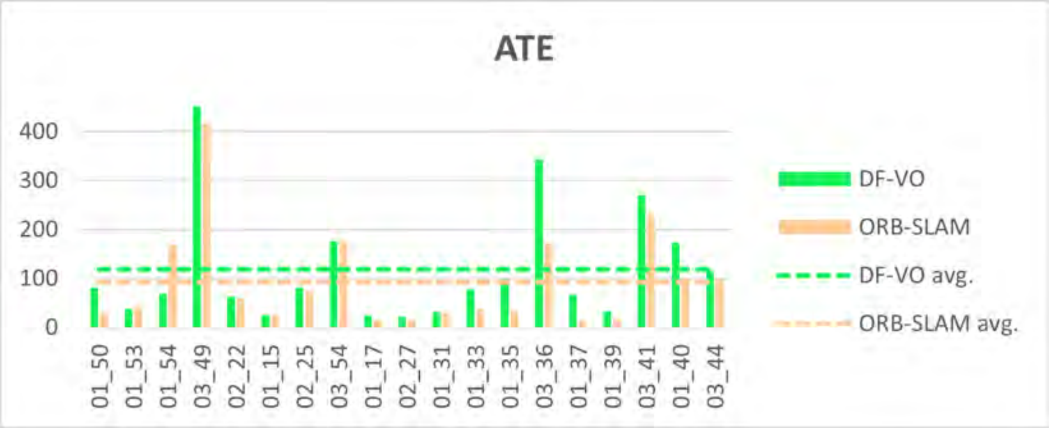


Figure 4.2.: ATE (m) of DF-VO and ORB-SLAM2 application on the generated proprietary *CAF* dataset. The most significant errors are estimated on the longest sequences (e.g., 03_49 and 03_36).

As shown in Figure 4.2, in most of the sequences the ATE is well below the average. However, in the longest sequences (e.g., 03_49, 03_36 and 03_41), the ATE increases greatly, which means that the average error is affected highly by those few long sequences. The ORB-SLAM2 pose estimation is 42% better than DF-VO estimations when comparing the ATE in this scenario.

Comparing the results to those obtained by the same algorithms in the KITTI dataset, in the case of the ATE, the error of DF-VO in the KITTI dataset is 6.872m while in the *CAF* dataset is 152.32m. For ORB-SLAM2, the ATE is 26.480m and 89.72m in the KITTI dataset and *CAF* dataset, respectively. The t_{err} is also lower for the KITTI dataset, increasing from 1.972% to 70.07% for DF-VO between the standard and the proprietary datasets. In the case of ORB-SLAM2, the t_{err} also increases from 8.074% to 58.34% from KITTI dataset to *CAF* dataset. Consequently, the evaluation of the algorithms' pose estimation in the *CAF* dataset shows that both algorithms' errors are higher than those found in the KITTI dataset. However, it can be seen that ORB-SLAM2 outperforms DF-VO in this challenging scenario.

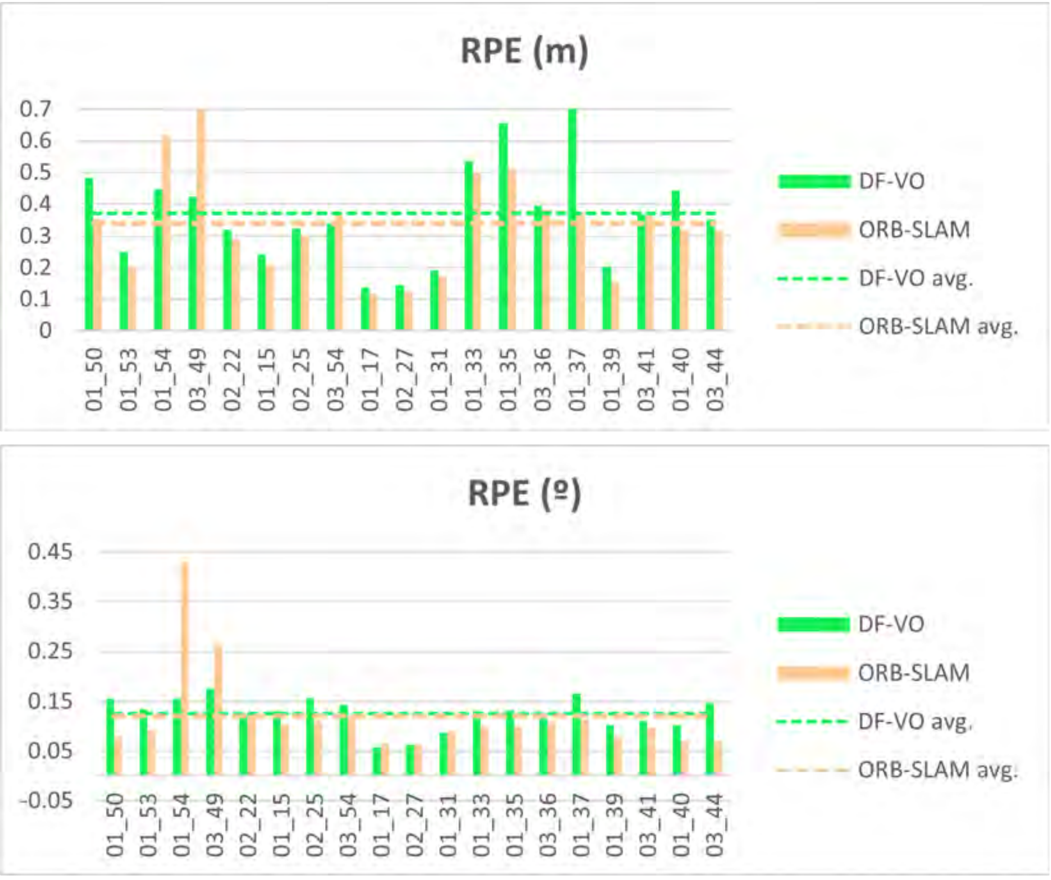


Figure 4.3.: Comparison of relative VO evaluation metrics when applying DF-VO and ORB-SLAM2 algorithms in CAF datasets. Translational and rotational components of relative errors are shown separately.

Algorithm	Metric	Avg. Err
DF-VO [133]	t_{err} (%)	1.972
	r_{err} (º/100m)	0.365
	ATE	6.872
	RPE (m)	0.041
	RPE (º)	0.038
ORB-SLAM2 (w/o Loop Closure) [91]	t_{err} (%)	8.074
	r_{err} (º/100m)	0.304
	ATE	26.480
	RPE (m)	0.130
	RPE (º)	0.063

Table 4.4.: Quantitative results of ORB-SLAM2 [91] and DF-VO [133] in the KITTI Odometry dataset. ORB-SLAM2 is executed without loop closure, as the sequences from *CAF* proprietary dataset do not contain loops.

Looking at the results obtained in the *CAF* dataset (resumed in Table 4.4, an additional experimentation was needed to ensure that the algorithms' performance was a results of the scenario characteristics. For that, the *CAF* sequences were shortened to just platform areas which have more similar lighting conditions of the KITTI dataset and, consequently, more limited lighting challenges.

Algorithm	Metric	Avg. Err
DF-VO [133]	t_{err} (%)	27.123
	r_{err} (°/100m)	6.824
	ATE	3.116
	RPE (m)	0.063
	RPE (°)	0.036
ORB-SLAM2 [91]	t_{err} (%)	14.011
	r_{err} (°/100m)	2.023
	ATE	2.945
	RPE (m)	0.04
	RPE (°)	0.029

Table 4.5.: Average standard VO errors in *CAF* dataset when reducing the sequences to platform areas without lighting challenges.

In this specific case, the errors are reduced to similar values (see Table 4.5) of executing DF-VO, and ORB-SLAM2 in the KITTI dataset [91], [133] (see Table 4.4). For instance, DF-VO achieves an RPE (m) of 0.027 in the KITTI dataset and 0.063 in the platform *CAF* dataset. ORB-SLAM2 achieves an ATE of 9.464 in KITTI dataset while 2.945 is achieved in the shortened sequences of *CAF* dataset. Seeing ATE values that are lower in the shortened *CAF* dataset than in the KITTI dataset could be related to the sequences' length, as these sequences are shorter than those in the KITTI dataset.

For the evaluation of these shortened sequences that are under 80m, the t_{err} and r_{err} have been calculated with sequence lengths of $\{10, 20, 30, 40, 50, 60, 70\}m$. The difference in the t_{err} obtained by DF-VO may be caused by the lack of training of the deep models under DF-VO or by the sequences length, as the authors of KITTI that propose these metrics mention, the error may be biased in short sequences.

These results seem to support that the challenging scene conditions hinder the application of VO algorithms in such scenarios. The qualitative VO/vSLAM results of DF-VO and ORB-SLAM2 on *CAF* dataset are shown in the following section (Section 5).

4.5 Conclusion

This chapter discusses the experimentation of applying VO/vSLAM algorithms in the *CAF* proprietary datasets (whole trajectory and platforms only). It can be seen that the results in the *CarDriving* dataset are similar to the results obtained in the standard KITTI dataset. However, the performance of reference DF-VO and ORB-SLAM2 algorithms is considerably degraded in the whole trajectory *CAF* dataset. ORB-SLAM2 shows to perform better than DF-VO in this scenario, obtaining lower errors in all the used metrics.

As mentioned in [211], geometry-based VO algorithms such as ORB-SLAM2 suffer a scale drift issue if additional mechanisms as loop detection are not used. This effect is increased when ideal visual conditions are not met. The results obtained in the *CAF* dataset raise the need of dealing with scale drift and low lighting related visual conditions. They require an adaptation procedure of geometric algorithms to handle scenarios that contain more challenging visual characteristics (low-light, low-textures, or non-lambertian surfaces).

In the case of DF-VO, being a hybrid algorithm, the estimation errors might be related to the geometric or learning parts of the algorithm. The geometric part follows the same issues faced by other geometric algorithms such as ORB-SLAM2. The estimation error of the learning part of the algorithm could be reduced by training the deep models (Monodepth2 [157] for depth estimation and LiteFlowNet [212] for flow estimation) in the target scenario. However, the authors of Monodepth2 mention that learning depth from such challenging scenarios is difficult for this specific network.

Nevertheless, these results obtained by the reference algorithms in the CAF dataset require an adaptation of reference VO solutions in order to become applicable in challenging scenarios, such as the underground railway domain.

5 Data enhancement for VO/vSLAM in an urban underground railway scenario

As seen in the results from previous sections, VO/V-SLAM algorithms performance is degraded when dealing with complex conditions of an urban underground railway scenario. This scenario presents certain difficulties compared to other standard scenarios, such as significant light changes from tunnel areas to platforms, insufficient illumination, non-Lambertian surfaces, and low textures in tunnels.

One of the main problems of these challenges is the poor lighting conditions, as it is related to all the others, and it is the base of them. Data enhancement has been considered in order to reduce the poor lighting effect in VO/V- SLAM results. The data enhancement process aims to handle the lighting-related limitations of state-of-the-art VO algorithms by reducing the lousy lighting conditions in the target domain. For that, an analysis of data augmentation techniques that can reduce the impact of poor lighting conditions has been carried out.

However, many VO/V-SLAM algorithms rely on camera calibration information for ego-motion estimation (e.g., [81], [85], [91], [133]). Camera calibration parameters are tied to image geometry, and a concern about the influence of this transformation in the calibration parameters of the camera raised. Consequently, an exploration of the data enhancement effects on calibration parameters was carried out.

In this section the application of previously selected state-of-art ORB-SLAM2 and DF-VO algorithms in images augmented with a GAN-based enhancement algorithm is considered and evaluated. Furthermore, as ORB-SLAM2 is a non-deterministic algorithm, the effect of the enhancement in the dispersion of the estimated poses among different executions is be examined.

This work has been presented on the IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics (ECSMS2021) in a paper called *Image enhancement using GANs for Monocular Visual Odometry*. In this paper, image enhancement techniques application is proposed an evaluated. It is shown that VO solutions performance is improved when applying those techniques.

This chapter is divided into the following sections. First, a literature review of data enhancement methods applied in the Computer Vision research works is made. Next, the enhanced datasets generation is explained. Then, a selected data enhancement algorithm from the literature review is applied in the proprietary datasets *CAF* and *CarDriving*. After that, as camera calibration parameters are fundamental for the state-of-the-art VO/vSLAM algorithms, the augmentation effect on camera calibration parameters is studied. Finally, the results of ORB-SLAM2 and DF-VO algorithms in generated enhanced dataset are shown and main conclusions are drawn.

5.1 Data enhancement in Computer Vision

The data manipulation is a very common strategy in the Computer Vision and Deep Learning research communities. Data enhancement in the computer vision field refers to techniques that improve the quality of images to enhance the processing of those images by computer vision algorithms [213]. Data enhancement sometimes is mixed up with Data augmentation. However, Data augmentation refers to increasing the data diversity and quantity for Deep Learning model training to improve the learning model inference performance.

Several enhancement methods can be found in the spatial domain, where pixel values are manipulated to achieve the desired enhancement. Given an image f, the data enhancement is defined as a transformation T that produces image g represented in equation 5.1,

$$g = T(f) \tag{5.1}$$

where each pixel value in f is mapped to a pixel in g. Following the taxonomy of image augmentation approaches in [214], data enhancement techniques can be divided into basic image manipulations or deep learning-based techniques.

Among the fundamental image transformations, changes in pixel values can be found such as color space transformation [215]–[218], kernel filters application [219], edge enhancement, noise injection [220], [221], or thresholding. Other primary augmentations are based on geometric operations, such as rotation, cropping, translation, scale, or flipping [222]. Another proposed technique is mixing images by averaging their pixel values [223], [224].

Deep Learning-based enhancement approaches started with Szegedy *et al.* [225] proposing *adversarial examples* to define small perturbations done to images to change deep networks predictions. Later, they were further studied by Goodfellow *et al.* [226]. From those works, later, Goodfellow *et al.* [227] proposed *Generative Adversarial Networks (GANs)*.

Most Deep Learning-based enhancement methods are GAN-based, and they consist of two neural networks competing with each other during their training. These networks are composed of a generative model that captures data distribution (often called latent variables) and a discriminative model that estimates the probability of

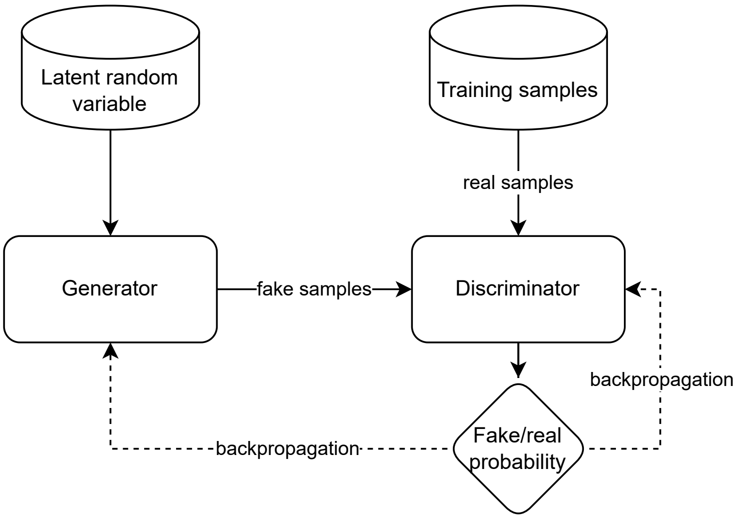


Figure 5.1.: GANs main architecture with generative and discriminative models and input data sources.

Lately, many research projects have focused on modifying the GAN framework with different network architectures, loss functions, or evolutionary methods [214]. For example, some research works have improved the capacity of GANs to create high-quality samples produce higher-resolution output images [228], [229].

Among these new architectures, Radford *et al.* presented DCGAN [230], an approach to increase the complexity of generator and discriminator models. Mirza *et al.* [231] proposed Conditional GANs that add a conditional vector to both deep models. Liu *et al.*. proposed Coupled GAN (CoGAN) [232], where an image can be translated from one domain to another (e.g., night images to daylight images) but without having corresponding images in those domains. Donahue *et al.* [233] proposed Bidirectional GAN (BiGAN), which included an encoder to make the mapping between domains bidirectional.

Later, other approaches to do an image-to-image translation were developed by Isola *et al.* (based on Conditional GANs) [234] or by Zhu *et al.* (CycleGAN) [229]. Following these works, Antioniu *et al.* [235] developed Data Augmentation Generative Adversarial Network (DAGAN) to generate new data from sample images. The generation is based on image conditional GANs, and these generated images are practically indistinguishable from real images.

As mentioned in [214], DCGANs, CycleGANs, and Conditional GANs seem to have the most significant potential for application in data enhancement. To summarize, several data enhancement methods are available depending on the research interest and existing dataset.

In this case, the research aimed to handle the poor visual conditions of the target scenario through data enhancement. The underground railway domain is characterized by varying lighting conditions (tunnel vs. platform) and low illumination (in tunnels), creating texture-less areas. Furthermore, the captured images are blurred as the train is in motion, degrading the visual perception. However, this research focuses only on enhancement methods that handle illumination issues.

Reviewing the literature, several methods have focused on lighting problems from different approaches such as deep curve estimation for low-light images [236], feature learning for image enhancement [237], synthetic dataset generation [238], or image decomposition into illumination and reflectance [239].

Jung *et al.* proposed Multi-Frame GAN (MFGAN) [240], an approach that introduces Generative Adversarial Networks (GAN) for VO with the introduction of a flow estimation-based loss term. It demonstrates the potential of GAN-based networks to enhance VO results in low-light conditions. In [241], LIME was presented, a low-light image enhancement method based on pixel values for each RGB channels. Another widely referenced GAN-based enhancement approach is *EnlightenGAN*, presented by Jian *et al.* [242].

EnlightenGAN is an unsupervised generative adversarial network for image lighting enhancement. It takes a dark image and enhances it lighting conditions using a encoder-decoder architecture. Then, global-local discriminators are used to minimize the adversarial loss, by evaluating randomly cropped local patches in addition to the image-level global discriminator. The deep model can improve the illumination of images and has adapted the weights of a VGG-16 model pre-trained on the ImageNet dataset [243]. *EnlightenGAN* has been trained with low-light and normal light images from several datasets released in [239], [244] and also High-Dynamic-Ranging (HDR) sources [245], [246]. It is the unique deep enhancement technique with the ability of making the training process with unpaired low/standard light images. Figure 5.2 illustrates the architecture of *EnlightenGAN* algorithm.

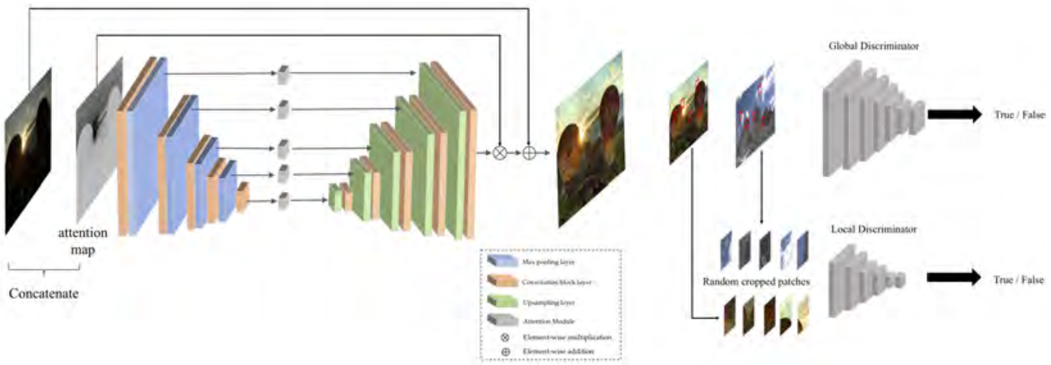


Figure 5.2.: EnlightenGAN architecture proposed by Jian *et al.* in [242]. The generator is composed by an attention-guided U-Net. Then, the generated images are introduced to global and local discriminators.

To summarize, this research has focused on the main degraded characteristics of the underground railway domain: significant light changes and low-light areas. These issues are related to scenes illumination, and consequently, the algorithms devoted to improving image lighting were aimed. From the SOTA, EnlightenGAN has been selected among the state-of-the-art data enhancement methods due to the simplicity and the results shown in other tasks.

5.2 Data enhancement effects on camera calibration parameters

Some VO/vSLAM algorithms are based on minimizing the reprojection error of consecutive frames captured by the camera (e.g., [86]). The error is estimated by solving the essential matrix, which depends on the intrinsic camera parameters, and assumes the camera satisfies the pinhole camera model. The camera parameters are obtained by a camera calibration process consisting on the minimization of the non-linear least-squares problem shown in [247].This process has a certain level of uncertainty that quantifies how reliable a calibration measurement is.

Accordingly, an evaluation of the selected algorithm impact on the camera calibration parameters and the related uncertainty values is required. It follows the idea of ensuring that the data enhancement methods do not affect the camera calibration parameters (e.g., the camera calibration parameters are still valid in the enhanced datasets).

5.2.1 Camera calibration setup

The effect of the EnlightenGAN enhancement technique in the calibration parameters of the proprietary datasets has been evaluated by analyzing the differences between the camera calibration parameters obtained in normal conditions and in enlightened conditions. Therefore, the focus was to check whether the calibration parameters do not change from non-enlightened images in standard illumination conditions to enlightened images in challenging conditions.

The calibration process has been carried out using Matlab's calibration tool *Stereo camera Calibrator*. It is based on the work presented by Zhang [248]. As they recommend, enough frames to cover the field of view, and to have a good distribution of different 3D orientations of the checkerboard are necessary (e.g., between 10 and 20 image pairs). Therefore, in this research, twenty frame pairs have been recorded for each calibration image set.

To analyze the changes in the calibration parameters obtained through the calibration process of the enlightening images, three image sets have been used. The *standard images* has been used to verify the calibration parameters given by the camera manufacturer and is composed by frames captured in appropriate light conditions. The *low-light images* are composed by frames recorded in poor lighting conditions so they can be enhanced to generate the third *enlightened images* set.

The effect of EnlightenGAN in the calibration parameters has been performed by comparing the calibration parameters obtained in the calibration process of the standard and the enlightened image sets. The scheme of the generated image sets is depicted in 5.3.

Table 5.1 resumes the calibration image sets generated for the calibration evaluation process. The defined image sets, the characteristics, and the frame quantity of each set are described.

In the figure 5.4, the calibration image set recording process setup is shown. It depicts the camera and a checkerboard calibration patterns positions. The calibration

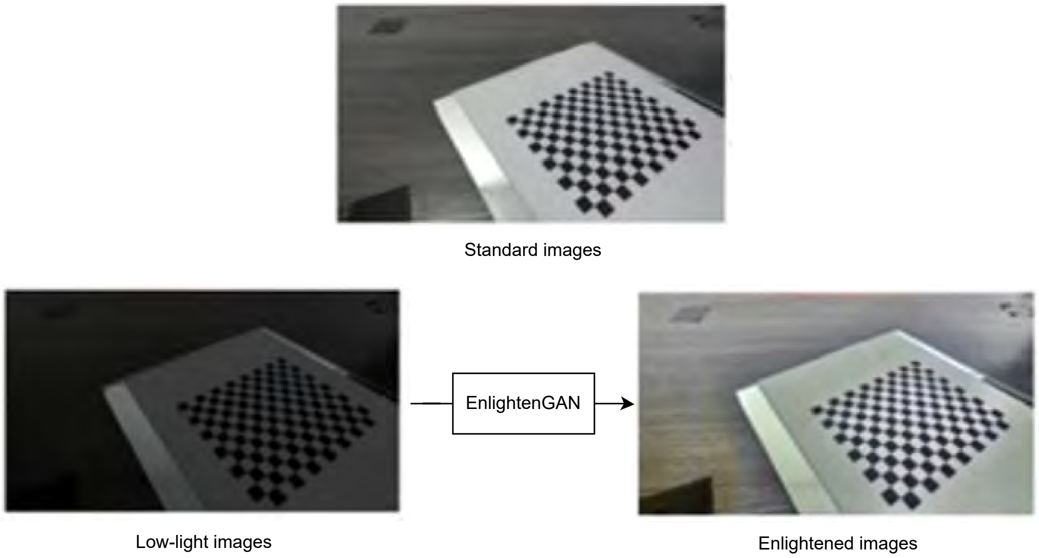


Figure 5.3.: Scheme of the image sets for the calibration verification. Enlightened images are obtained applying EnlightenGAN [242] to low-light images.

Image set	Frame pairs	Characteristics
Standard images	20	Images recorded in normal lighting conditions.
Low-light images	20	Images recorded with poor lighting conditions.
Enlightened images	20	Low-light images enhanced with the EnlightenGAN [242] algorithm.

Table 5.1.: Generated image sets for calibration using a camera and the EnlightenGAN [242] image enhancement technique. The characteristics and the frame pair quantity of each set are described.

pattern was placed in different poses inside the camera's visual area. Two images were taken in each pose: one with the light of the room switched on (for the *standard images* set) and the second with the light turned off, simulating poor lighting conditions (for the *low-light images* set). Therefore, the only change between a standard and a low-light image is the illumination.

5.2.2 Camera calibration evaluation

The calibration was evaluated using two standard metrics used in camera calibration experimentation processes: *Mean Reprojection Error (MRE)* and *uncertainty*. The MRE provides a qualitative measure of the accuracy during calibration. The reprojection error is the error in the distance between a pattern keypoint detected in an

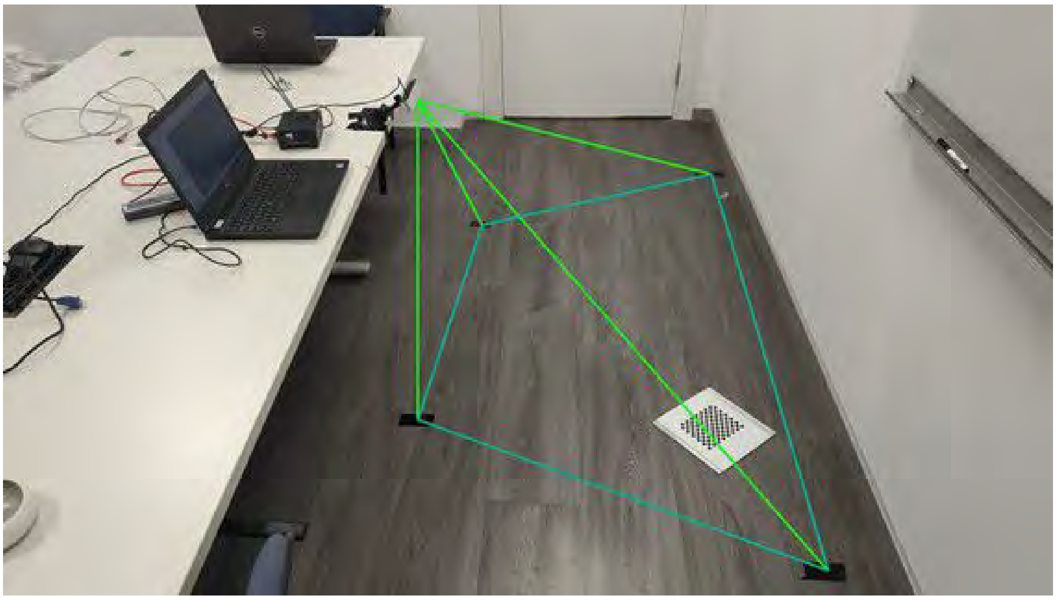


Figure 5.4.: Calibration dataset generation setup with the camera, the image recording area and the checkerboard pattern for calibration.

image used in the calibration and a corresponding world point projected into the same image. See the MRE definition in Equation 5.2:

$$MRE := \frac{1}{n * m} \sum_{i=1}^{n} \sum_{j=1}^{m} \sqrt[2]{(KP_{ij}\vec{p_i} - \vec{x_{ij}})^2} \tag{5.2}$$

where K is the intrinsic camera parameter matrix, P_{ij} is the camera pose, $\vec{p_i}$ are feature locations in 3D, and $\vec{x_{ij}}$ corresponds to their projection in the 2D image plane after correcting lens distortions.

The *uncertainty* represents the standard error corresponding to each estimated camera parameter and measures the dispersion of sample means around the populations mean [249]. In other words, quantifies all the possible differences between a measure and its real value, taking into account that a result of a measure is an approximation of its real value. The resulting uncertainty error can be used to calculate the confidence intervals. It is necessary to know the uncertainty to have a more realistic estimate, as the lower the uncertainty, the better the estimation.

Both the MRE and the uncertainty have been calculated with Matlab's calibration tool. The following tables represent the optimal calibration parameters given by the manufacturer, and the calibration results obtained in the different image sets. The MRE and the uncertainty of the main camera calibration parameters are measured: the focal length, the principal point, and the radial distortion.

Metric	Optimal calibration values
Focal length (f_u/f_v)	700.57 / 700.57
Principal point (c_x/c_y)	647.99 / 371.86
Radial distortion $(k1/k2)$	0.025 / -0.17

Table 5.2.: Optimal calibration values obtained from the camera manufacturer.

As shown in Table 5.3, the focal length and principal point estimated in the *standard images* set and the intrinsic camera parameters given by the camera manufacturer (see Table 5.2) are similar.

Metric	Standard images
MRE	0.1089
Focal length (f_u/f_v)	690.31 ± 4.171 / 689.58 ± 4.104
Principal point (c_x/c_y)	644.96 ± 0.546 / 381.21 ± 2.367
Radial distortion $(k1/k2)$	0.014 ± 0.002 / -0.013 ± 0.002

Table 5.3.: Standard images calibration results. The MRE and the estimated camera intrinsic parameters with their uncertainty are shown.

Table 5.4 shows the calibration results on *low-light images* and *enlightened images*. The results reveal that the focal length and principal point in both *low-light images* and *enlightened images* are also similar, being the mean reprojection error and uncertainties slightly higher in the *enlightened images* set.

Metric	Low-light images	Enlightened images
MRE	0.0775	0.0904
Focal length (f_u/f_v)	694.058 ± 2.558 / 693.605 ± 2.482	693.379 ± 3.204 / 692.858 ± 3.130
Principal point (c_x/c_y)	645.006 ± 0.351 / 380.768 ± 1.490	644.871 ± 0.43 / 380.984 ± 1.840
Radial distortion $(k1/k2)$	0.015 ± 0.001 / -0.017 ± 0.001	0.015 ± 0.001 / -0.017 ± 0.002

Table 5.4.: Calibration results in low-light images and enlightened images. The MRE and the estimated camera intrinsic parameters with their uncertainty are shown.

The results show that *EnlightenGAN* algorithm slightly increases the MRE and the calibration parameters' uncertainty. However, even the MRE and the uncertainty of the camera's intrinsic parameters being slightly worse than the optimal calibration parameters provided by the manufacturer (see Table 5.2), they are still similar, and, consequently, it can be concluded that *EnlightenGAN* algorithm does not disturb the camera calibration parameters significantly.

Consequently, the *EnlightenGAN* algorithm can be used without having to consider camera calibration afterwards, and it can be integrated in any VO/vSLAM algorithm directly.

5.3 Enhanced datasets generation: *EnlightenCarDriving* and *EnlightenCAF*

Once the enhancement algorithm has been selected and the effect on camera calibration parameters has been verified, the generation of the enhanced datasets has been pursued. Figure 5.5 shows the diagram of the enhancing procedure.

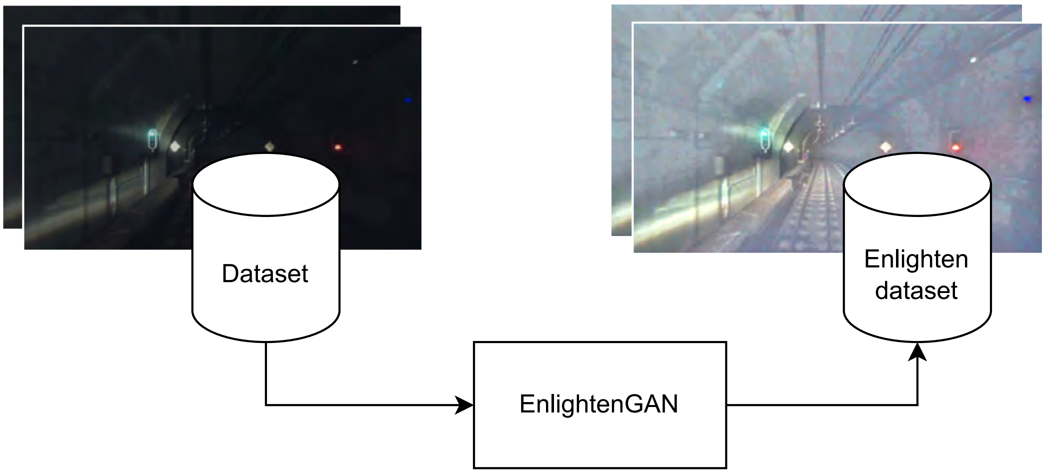


Figure 5.5.: Procedure followed for dataset enhancement through EnlightenGAN algorithm.

The *EnlightenGAN* algorithm implementation for inference and weights for the learning models have been taken from [242].

Given the two generated proprietary datasets (*SteroDriving* and *CAF*, see Chapter 3 for these datasets generation), *EnlightenCarDriving* enhanced dataset was generated from the *DrivingCar* dataset, and *EnlightenCAF* from the proprietary *CAF* dataset.

In the case of *DrivingCar* dataset, the low-light sequences have been enlightened to generate enhanced sequences 01 and 03 from *EnlightenCarDriving*.

For the *CAF* dataset, all the sequences have been enlightened to generate a new dataset where all the frames' light conditions are enhanced. Figure 5.6 shows the result of the enhancement in sample images from both datasets (left-side image is the original one, right-side image is the enhanced).

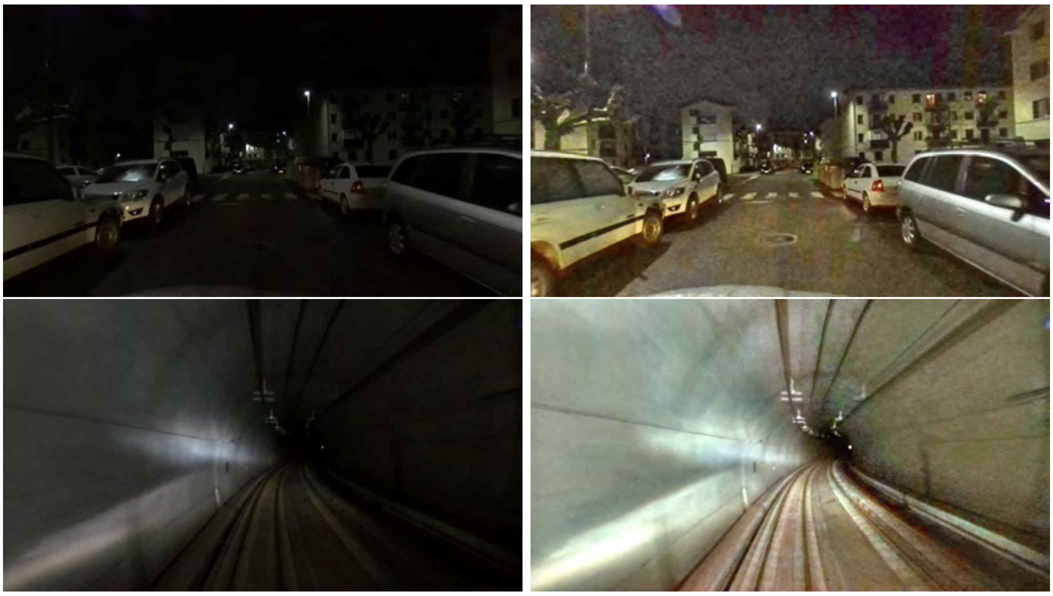


Figure 5.6.: Result of the enhancement of a frame sample for *CarDriving* and *CAF* datasets, respectively. The right-side frame is the enlightened version of the left-side frame. *EnlightenGAN* algorithm is applied to the enhanced samples.

5.4 DF-VO in *EnlightenCarDriving* dataset

This section resumes the performance achieved by DF-VO [133] in the enhanced *EnlightenCarDriving* dataset. This dataset is generated enlightening the low-light sequences from *CarDriving* dataset (sequences 01 and 03). The recordings were done driving a car through urban scenarios by night. The ground truth data was generated using the geometric VO/vSLAM algorithm ORB-SLAM2 [91] and corrected utilizing the traveled distance through a GPS. ORB-SLAM2 has become an state-of-the-art algorithm for the VO/vSLAM research community, and has been previously used as the ground truth data generation algorithm [106]. The dataset generation process is more deeply depicted in Section 3.3.1.

The performance of DF-VO in *EnlightenCarDriving* has been evaluated using the same metrics from the previous section (Average Absolute Trajectory Error – ATE and Relative Pose Error – RPE). The results are summarized in Table 5.5.

The results show that the minimum ATE was obtained in the enlightened sequence 01 (the Aragoa trajectory), 4.88m over 6.99m of the enlightened sequence 03 (Musakola trajectory). The RPE follows the same behavior in both sequences. For the average translational error (t_{err}), a lower pose estimation error is obtained in the Aragoa trajectory (7.20% over 9.82%).

Trajectory	Aragoa	Musakola	Avg. Err.
Seq.	01	03	
t_{err} (%)	7.20	9.82	8.51
r_{err} (°/100m)	2.51	4.23	3.37
ATE	4.88	6.99	5.935
RPE (m)	0.087	0.318	0.2025
RPE (°)	0.183	0.398	0.2905

Table 5.5.: DF-VO evaluation in *EnlightenCarDriving* dataset generated from enhancing *CarDriving* dataset through EnlightenGAN algorithm. Sequence 01 belongs to the Aragoa trajectory and sequence 03 to the Musakola trajectory.

The difference in the results of the two sequences could be related to the trajectories lengths, as the Aragoa trajectory (sequences 00 and 01) is 300m in length, while the Musakola trajectory (sequences 02 and 03) is 450m in length. DF-VO is an incremental algorithm, the poses are estimated based on the previous estimated poses and the error is accumulated over time. Consequently, the pose estimation errors are more significant in longer trajectories.

5.4.1 Comparison of DF-VO in *CarDriving* and *EnlightenCarDriving* datasets

This section depicts the comparison between the results obtained in the *CarDriving* proprietary dataset and the enhanced *EnlightenCarDriving* dataset. Table 5.6 shows the errors of applying DF-VO in *CarDriving* (only in the night sequences 01 and 03) and *EnlightenCarDriving* (the enlightened sequences 01 and 03) datasets.

Trajectory	Aragoa		Musakola	
Seq.	CD	ECD	CD	ECD
t_{err} (%)	8.26	7.20	12.52	9.82
r_{err} (°/100m)	4.36	2.51	5.92	4.23
ATE	5.71	4.88	8.73	6.99
RPE (m)	0.268	0.087	0.38	0.35
RPE (°)	0.784	0.183	0.565	0.46

* CD=CarDriving, ECD=EnlightenCarDriving

Table 5.6.: DF-VO evaluation in *CarDriving* (night sequences) and *EnlightenCarDriving* datasets (sequences 01 and 03). Standard metrics from VO/vSLAM research community are used. The results in the two trajectories are shown for each dataset.

ATE over a non-enlightened 300m run from the sequence 01 of Aragoa is 5.71m. However, the enlightened sequence has an ATE of 4.88m. Therefore, the EnlightenGAN enhancement algorithm has improved the results obtained in the original sequence by 1.17m. The same behavior can be appreciated for the Musakola trajectory (improvement of 1.74m in the enlightened sequence). The average translational error (t_{err}) is also reduced from 8.26% to 7.20% for Aragoa trajectory, and from 12.52% to 9.82% for Musakola trajectory.

The relative errors (RPE) are reduced in the enlightened sequences when compared to the sequences recorded by night. Translational RPE is reduced in 0.181m for Aragoa trajectory and 0.03m for Musakola trajectory.

Manifestly, the experimental results show that *EnlightenGAN* improves the DF-VO performance in low-light car scenarios. Seems that EnlightenGAN is capable of increasing the performance of DF-VO by improving the deep flow or depth estimations that contribute to the pose calculation through 2D-2D or 3D-2D correspondences.

5.5 DF-VO and ORB-SLAM2 in *EnlightenCAF* dataset

This section resumes the performance of the state-of-the-art VO/vSLAM algorithms ORB-SLAM2 and DF-VO in an enlightened urban underground railway scenario. Then a comparative between the performance of both algorithms in the enhanced and non-enhanced datasets (*CAF* and *EnlightenCAF*) is performed.

The enhanced *EnlightenCAF* dataset was generated through enlightening the *CAF* dataset. This dataset contains 19 image sequences of a train driving trough the L3 railway in Bilbao. The ground truth was generated through a novel method that synchronizes geodetic coordinates that represent the track positions, ATP monitored data from train sensors and a gradient map provided by the railway constructor. The dataset definition is explained more in depth in section 3.3.2).

Results are depicted in Table 5.7 and in the Figures 5.7, 5.9, and 5.8. The average translational (t_{err}) and rotational (r_{err}) errors for ORB-SLAM2 and DF-VO in *EnlightenCAF* dataset is shown in Figure 5.7. It can be noticed that the algorithms' performance continues the same behavior as in the non-enhanced datasets. ORB-SLAM2 outperforms DF-VO in both metrics. The highest t_{err} can be found in sequence *01_35*, a situation that might be produced because it is the sequence that has the steepest turn. In general, the average errors (t_{err} and r_{err}) measured in the performance of both algorithms continue to be higher than the results obtained in the standard KITTI dataset.

Algorithm	Record	14_11_2021 (->Matiko)									
	Seq	01_50	01_53	01_54	03_49	02_22	01_15	02_25	03_54	01_17	02_27
DF-VO	t_{err} (%)	65.47	50.36	55.66	86.61	55.76	54.15	52.88	81.31	52.55	49.88
	r_{err} (°/100m)	6.82	20.46	18.43	13.04	19.98	21.86	23.2	23.01	14.29	15.09
	ATE (m)	145.45	95.03	112.02	410.95	142.77	54.1	110.67	248.93	20.59	31.22
	RPE (m)	0.389	0.211	0.313	0.421	0.254	0.25	0.319	0.386	0.099	0.097
	RPE (°)	0.101	0.103	0.124	0.143	0.114	0.125	0.132	0.132	0.052	0.056
ORB-SLAM2	t_{err} (%)	38.91	35.39	47.02	79.83	29.66	43.49	46.41	88.09	32.41	31.69
	r_{err} (°/100m)	8.3	13.13	12.38	10.71	13.83	15.41	16.19	17.58	8.19	8.14
	ATE (m)	44.52	44.67	59.91	245.68	53.47	27.68	73.42	188.73	13.93	16.37
	RPE (m)	0.35	0.193	0.271	0.511	0.239	0.216	0.309	0.416	0.084	0.084
	RPE (°)	0.089	0.099	0.099	0.095	0.109	0.113	0.112	0.112	0.066	0.066

Algorithm	Record	14_11_2021 (->Kukullga)									Avg. Err.
	Seq	01_31	01_33	01_35	03_36	01_37	01_39	03_41	01_40	03_44	
DF-VO	t_{err} (%)	38.39	76.15	120.51	68.15	71.65	57.82	83.49	79.32	75.66	67.1457
	r_{err} (°/100m)	9.02	20.94	20.7	15.99	23.72	20.47	16.01	27.03	24.14	18.6421
	ATE (m)	36.08	101.08	113.25	354.11	139.69	59.31	268.09	247.83	191.44	151.7163
	RPE (m)	0.151	0.386	0.568	0.398	0.444	0.257	0.415	0.326	0.348	0.3174
	RPE (°)	0.079	0.103	0.132	0.107	0.142	0.11	0.105	0.099	0.084	0.1075
ORB-SLAM2	t_{err} (%)	33.16	80.1	121.26	69.55	60.23	48.16	93.96	56.69	56.73	**57.5126**
	r_{err} (°/100m)	7.35	16.3	17.57	14.53	15.1	13.48	11.74	10.62	10.69	**12.6968**
	ATE (m)	30.95	37.4	91.2	176.23	96.92	20.07	228.37	99.8	96.98	**86.6473**
	RPE (m)	0.13	0.39	0.618	0.38	0.438	0.257	0.414	0.304	0.303	**0.3108**
	RPE (°)	0.093	0.096	0.112	0.109	0.089	0.088	0.109	0.076	0.074	**0.0950**

Table 5.7.: DF-VO and ORB-SLAM2 application evaluation in the EnlightenCAF dataset using standard VO evaluation metrics: Average Translational Error (t_{err}), Average Rotational Error (r_{err}), ATE and RPE. The sequences are organized by the direction they are recorded. The average errors for all 19 sequences are calculated, and the best result is in bold.

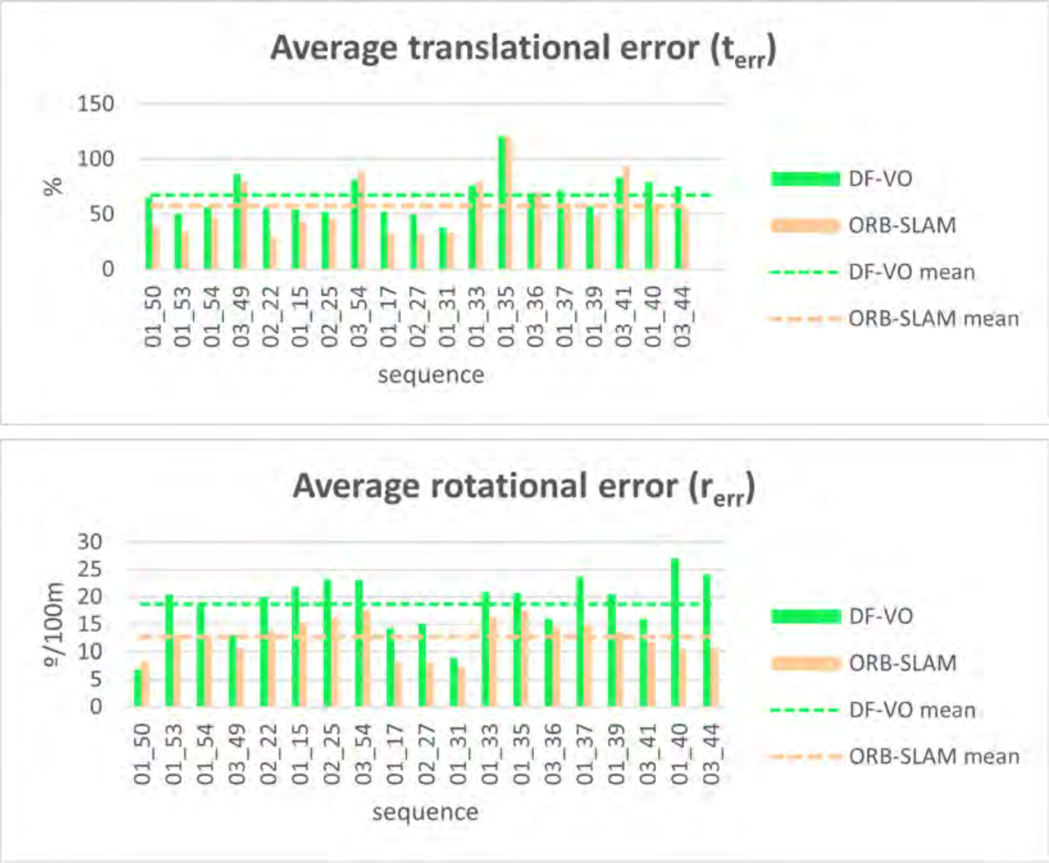


Figure 5.7.: Comparison of the average translational (t_{err}) and rotational (r_{err}) errors when applying DF-VO and ORB-SLAM2 algorithms in *EnlightenCAF* dataset.

As shown in Figure 5.8, the longest sequences (e.g., sequences *03_49*, *03_36* or *03_41*) are the ones that have the higher ATE error for both algorithms. The mean ATE error is 151.71m for DF-VO, while it is 86.64m for ORB-SLAM2. The higher ATE error is obtained in the sequence *03_49* (410.95m over a trajectory of 3145m) by DF-VO algorithm. This sequence (as the other long sequences) mixes tunnel and platform areas, with more low-light frames and big light changes between the two underground environments.

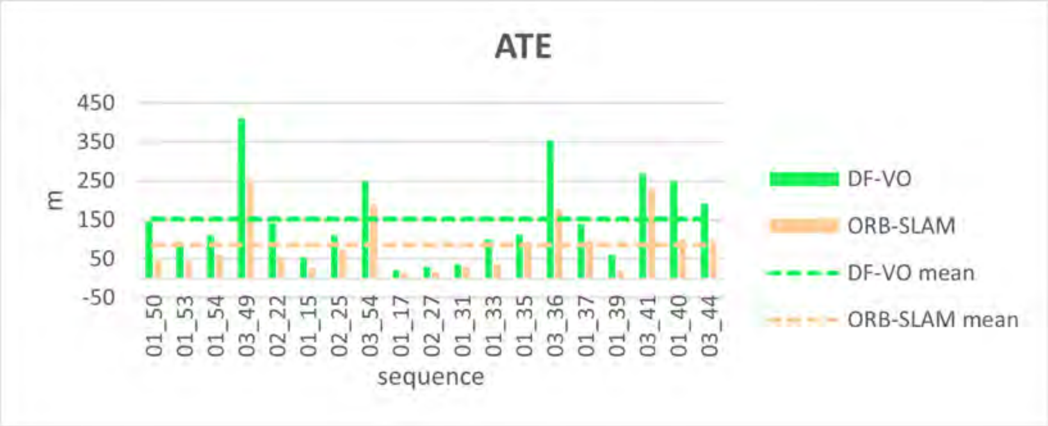


Figure 5.8.: ATE obtained by DF-VO and ORB-SLAM2 algorithms on the enhanced *EnlightenCAF* dataset. The mean error for each algorithms is also shown.

In the case of the RPE, also depicted in Figure 5.9, it can be seen that ORB-SLAM2 also obtains better results than DF-VO. However, the mean translational RPE of all the sequences is similar between both algorithms (0.3174m for DF-VO and 0.3108 for ORB-SLAM2). The minimum translational RPE is obtained in sequences *01_17* and *02_27* (0.084m), which belong to the same trajectory, by ORB-SLAM2. DF-VO also obtains its lowest estimation errors in those two sequences. Those sequences are characterized by a nearly linear path with almost no curves. ORB-SLAM2 obtains the maximum RPE in the sequence *01_35* (0.618m). In some of the sequences, the translational RPE is similar between DF-VO and ORB-SLAM2 when estimating the poses. For example, in the sequence *01_33*, a translational RPE of 0.386 and 0.39 is obtained by DF-VO and ORB-SLAM2, respectively.

Consequently, the results show that the algorithms' performance in enhanced *EnlightenCAF* dataset is improved with respect to the non-enlightened *CAF* dataset. The behavior observed in the *CarDriving* and *EnlightenCarDriving* datasets is preserved. Furthermore, although the pose estimation errors have decreased generally, the results of both algorithms continue to be higher than those obtained in the KITTI dataset. Also, ORB-SLAM2 outperforms DF-VO in this scenario even in the enhanced images.

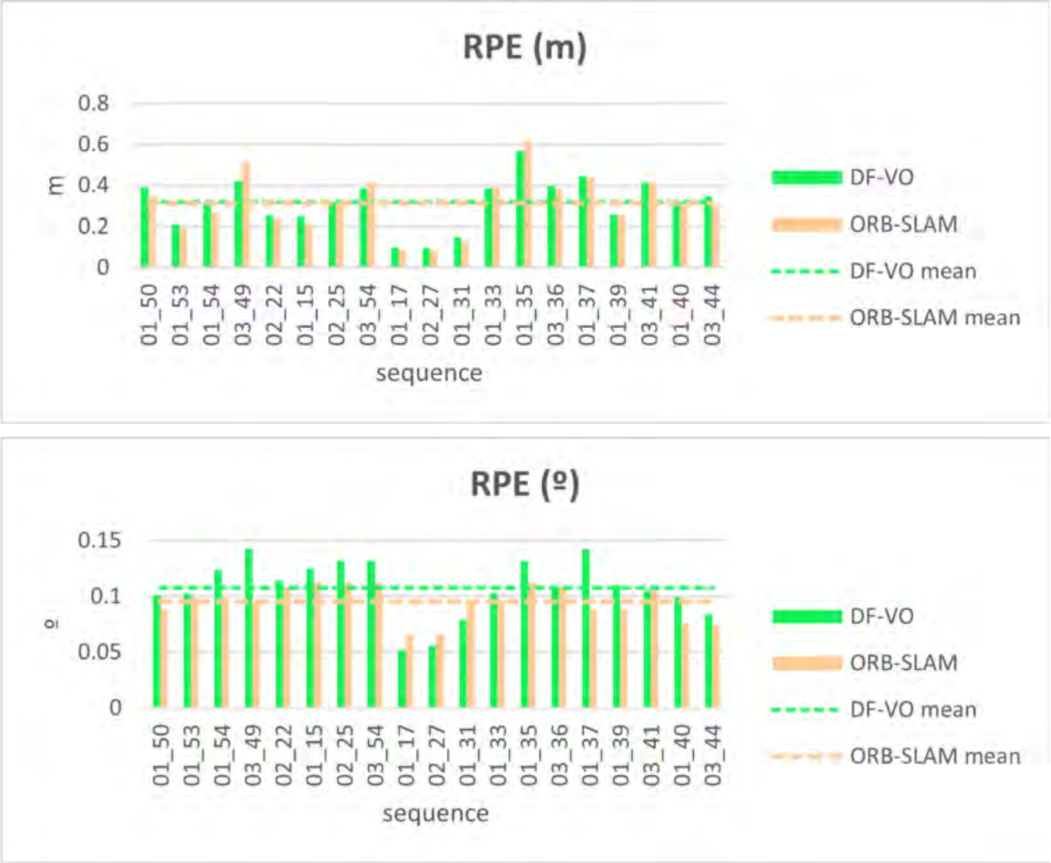


Figure 5.9.: Comparison of relative VO evaluation metrics when applying DF-VO and ORB-SLAM2 algorithms in *EnlightenCAF* datasets. Translational and rotational components of relative errors are shown separately.

5.5.1 Comparison of DF-VO and ORB-SLAM2 in *CAF* and *EnlightenCAF* datasets

The comparative between the algorithms in *CAF* and *EnlightenCAF* datasets shows that EnlightenGAN reduces the ego-motion estimation errors for DF-VO and ORB-SLAM2 algorithms. The following figures demonstrate the reduction in the mean t_{err}, r_{err}, ATE, and RPE for all the sequences in *EnlightenCAF* for both algorithms.

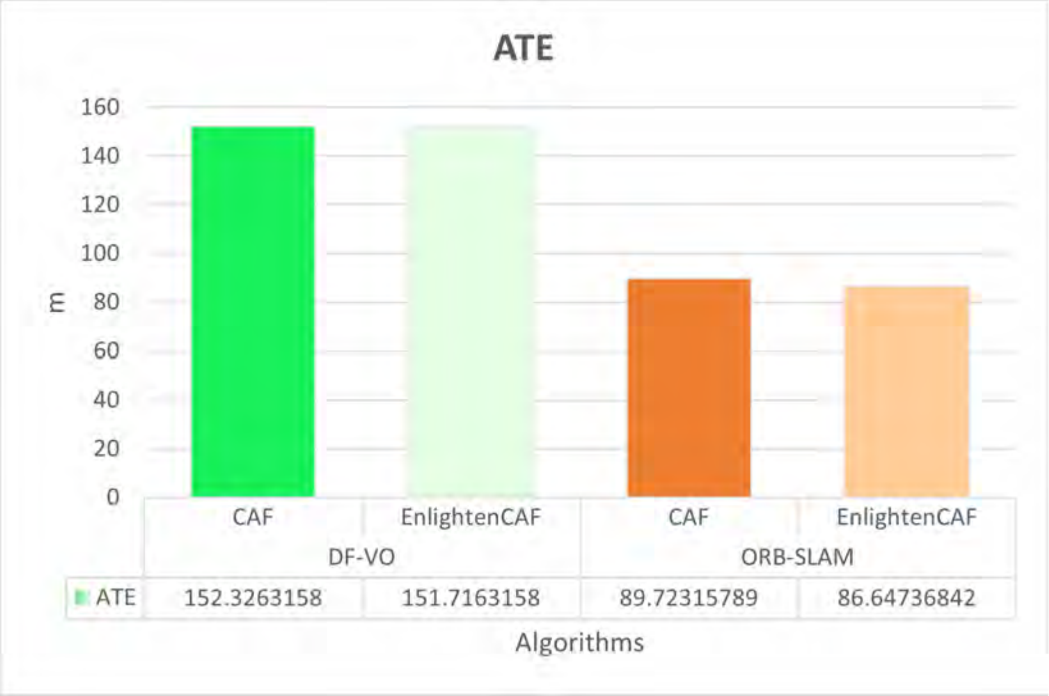


Figure 5.10.: Comparative of ATE error of DF-VO (in green) and ORB-SLAM2 (in orange) algorithms in *CAF* and *EnlightenCAF* datasets. For each dataset, the mean ATE error is shown. The error is reduced for the enlightened sequences.

ATE, shown in Figure 5.10, reduces by 0.40% for DF-VO estimation and by 3.43% for ORB-SLAM2 estimation with respect to *CAF* dataset results. This seems to point out what has been theorized previously, that low-lighting conditions affect the performance of VO/vSLAM algorithms and *EnlightenGAN* reduces that effect. Figure 5.11 shows RPE behavior when selected VO algorithms are applied in EnlightenCAF dataset.

In the case of the RPE (see Figure 5.11), the DF-VO estimation is improved by 3.01% and by 28.19% for translation and rotation components, respectively. ORB-SLAM2 estimation improves by 1.96% for the RPE translation component and 0.28% for the rotation component. The use *EnlightenGAN* use improves the DF-VO performance more than the ORB-SLAM2 performance when analyzing the RPE.

Figure 5.12 resumes the comparison of the average translational (t_{err}) and rotational (r_{err}) errors obtained by DF-VO and ORB-SLAM2 in the *CAF* and *EnlightenCAF* datasets. t_{err} and r_{err} are reduced in 4.18% and 1.64% for DF-VO, and in 1.43% and 0.44% for ORB-SLAM2. As with the other metrics, the errors of the learning-based DF-VO are decreased more than the errors obtained by the estimations of ORB-SLAM2.

Table 5.8 resumes the comparison between the average errors in CAF and EnlightenCAF datasets. The results show that the use of EnlightenGAN enhancement in low-light images improves the performance of the state-of-the-art VO/vSLAM algorithms.

Algorithm	Metric	CAF	ECAF
DF-VO	t_{err} (%)	70.0726	67.1457
	r_{err} (°/100m)	18.9521	18.6421
	ATE (m)	152.3263	151.7163
	RPE (m)	0.3273	0.3174
	RPE (°)	0.1497	0.1075
ORB-SLAM2	t_{err} (%)	58.3468	57.5126
	r_{err} (°/100m)	12.7531	12.6968
	ATE (m)	89.7231	86.6473
	RPE (m)	0.3171	0.3108
	RPE (°)	0.0953	0.0950

* ECAF=EnlightenCAF

Table 5.8.: Average errors of DF-VO and ORB-SLAM2 evaluation in *CAF* and *EnlightenCAF* (ECAF) datasets. Standard VO/vSLAM evaluation metrics are used (Average translational and rotational errors, ATE and RPE).

From Table 5.8, it can be seen that the performance of both algorithms is improved in the enhanced sequences. An average ATE of 86.64m is obtained in the *EnlightenCAF* dataset with ORB-SLAM2, while DF-VO obtains an average ATE of 151.71m. To understand the magnitude of that error, the average length of the sequences in the generated *CAF* and *EnlighenCAF* datasets is 2343.34m. The average translational (t_{err}) and rotational (r_{err}) errors are also reduced in the enhanced sequences. From 58.34% to 57.51% for the ORB-SLAM2 algorithm and 70.07% to 67.14% for DF-VO. The t_{err} and the r_{err} are reduced by 2.80% and 1.04% on average between the two algorithms.

Although the pose estimation errors are high for both algorithms in this challenging scenario, EnlightenGAN reduces those errors more for the DF-VO algorithm than for the ORB-SLAM2 algorithm. However, ORB-SLAM2 still has more accurate pose estimation results.

Figure 5.13 shows a qualitative result comparison of DF-VO and ORB-SLAM2 in CAF and EnlightenCAF datasets, where the estimation of some sequences are compared to reference data trajectories. As in the original trajectories (*CAF dataset*), it can be seen that the algorithms can estimate the shape of the *EnlightenCAF* trajectories. However, a scale underestimation problem appears again. Furthermore, DF-VO results show that the rotation estimation is affected in the *EnlightenCAF* dataset.

In most sequences, an underestimation of scale is apparent for both algorithms, especially for DF-VO. The scale of DF-VO depends on the correspondences found in the flow estimation, and those correspondences are affected by the lighting conditions of tunnels in the underground environment. Also, a shift in estimating the curves made by the train during its travel is observed. For example, in the sequence *02_22*, DF-VO has issues detecting the correct curves once the train starts to turn to the left. ORB-SLAM2 estimations are closer to the ground truth trajectory. Overall, although a scale shift appears and curves are wrongly estimated, the shapes of the estimated trajectories are similar to the reference trajectories.

In conclusion, the results demonstrate that EnlightenGAN improves VO/vSLAM algorithms performance in the underground railway domain. Furthermore, the errors are more reduced for the learning-based DF-VO algorithm than for the geometric-based ORB-SLAM2 algorithm. It seems that the learning algorithms are more affected by the lighting conditions of the scenario. As in the *CAF* dataset, an influence of lighting conditions of the scenario can still be appreciated. This effect could be related to scale underestimation problems found in both algorithms, especially in the hybrid DF-VO.

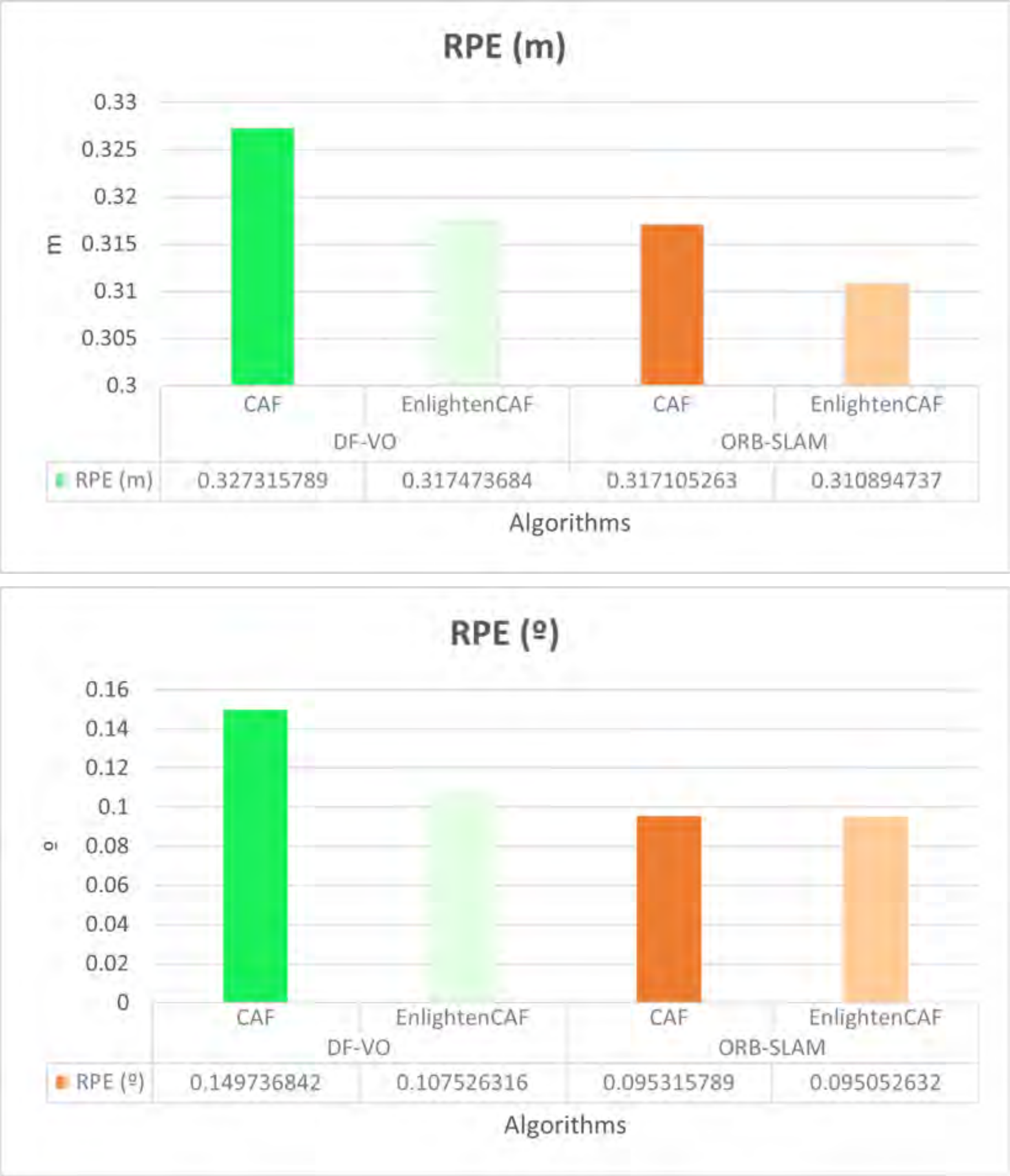


Figure 5.11.: Comparison of RPE when applying DF-VO and ORB-SLAM2 algorithms in *CAF* and *EnlightenCAF* datasets. Translational and rotational components of RPE are shown separately.

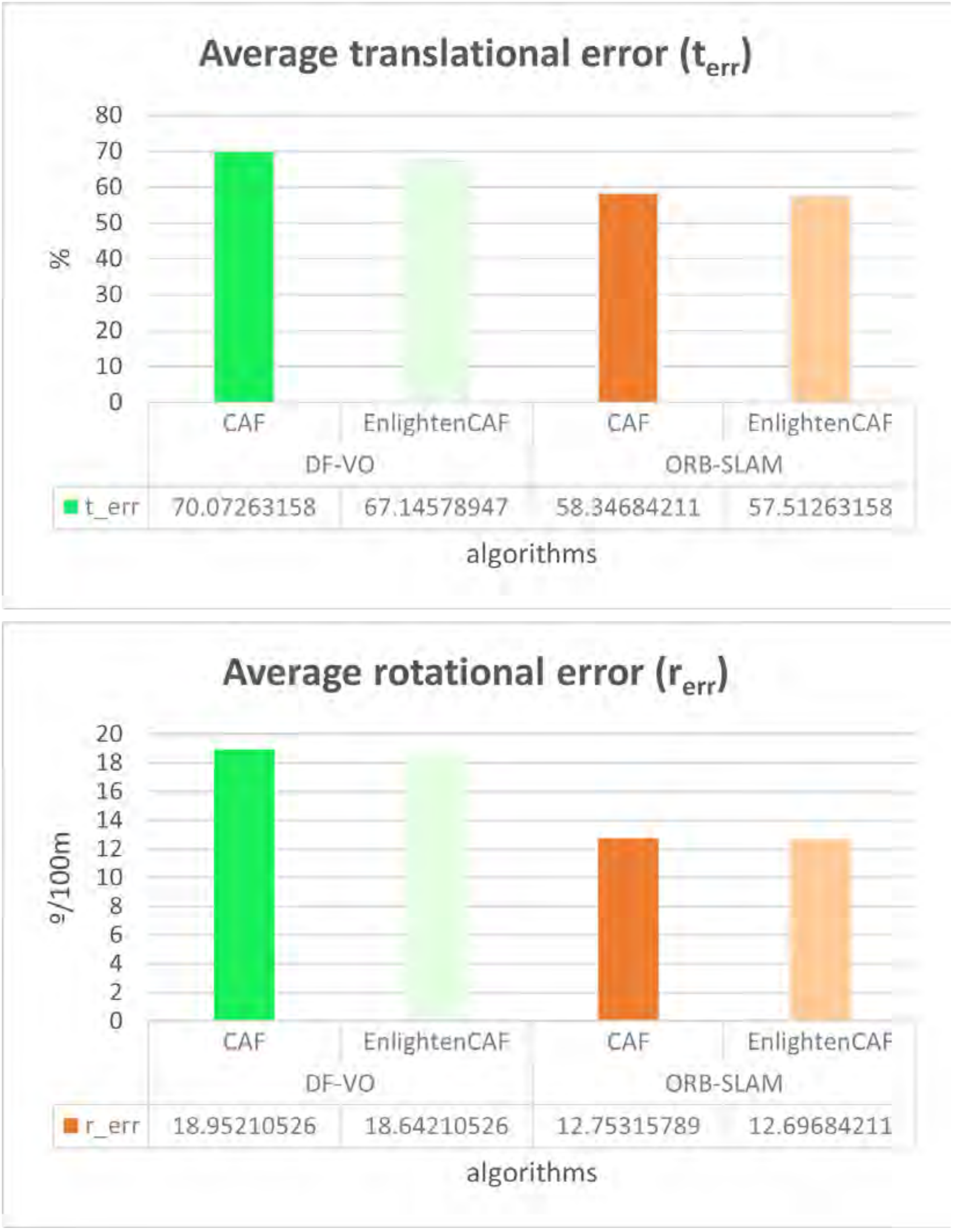
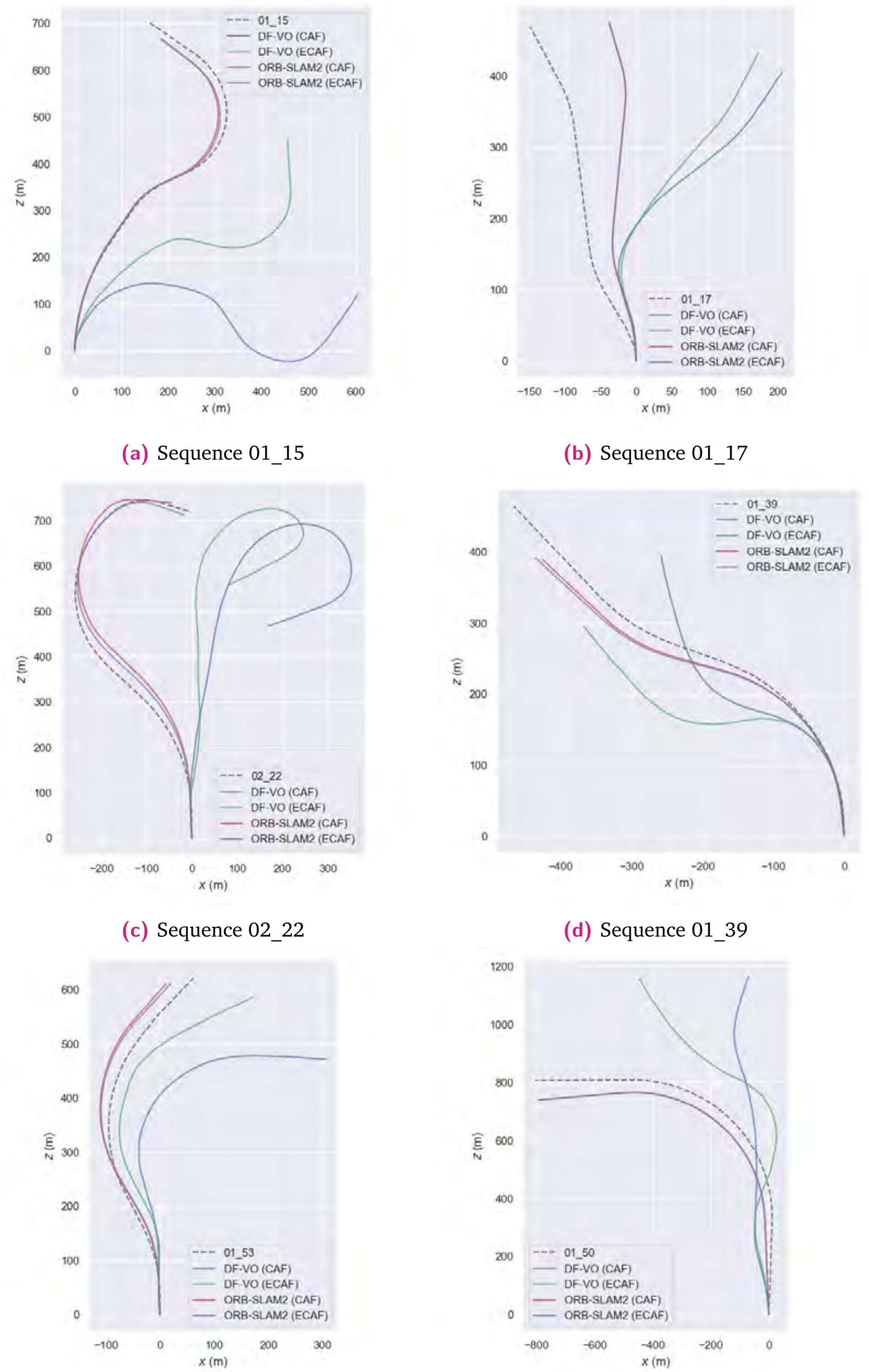


Figure 5.12.: Comparison of the average translational (t_{err}) and rotational (r_{err}) errors when applying DF-VO and ORB-SLAM2 algorithms in *CAF* and *EnlightenCAF* datasets. Translational and rotational components are shown separately. These errors measure the performance of the algorithms in different length subsequences.



(a) Sequence 01_15

(b) Sequence 01_17

(c) Sequence 02_22

(d) Sequence 01_39

(e) Sequence 01_53

(f) Sequence 01_50

Figure 5.13.: Comparison of ORB-SLAM2 and DF-VO application in some sample sequences in both *CAF* and *EnlightenCAF* (ECAF) datasets and the reference ground truth for each trajectory. The improvement from the non-enhanced to enlightened sequences can be appreciated more in the DF-VO algorithm estimations.

5.5.2 ORB-SLAM2 dispersion in enhanced EnlightenCAF dataset

As stated before, ORB-SLAM2 is a non-deterministic algorithm, where each execution is different because of the initialization process based on the reliability of descriptors matching. Therefore, the poses estimated by ORB-SLAM2 fluctuate among different runs. When evaluating the ORB-SLAM2 pose estimation, this fluctuation seems to be reduced when enhancing the frames with EnlightenGAN algorithm. Accordingly, an initial evaluation of ORB-SLAM2 dispersion has been done, where the impact of the enhancing EnlightenGAN in that dispersion is observed.

The dispersion of poses among different executions has been evaluated using standard metrics [250]. These metrics include the *variance* (σ^2) and the *Coefficient of Variation* (cv). The variance measures the variability of the values from the sample's mean. σ^2 is defined in the following equation 5.3:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} \tag{5.3}$$

where x_i is the i^{th} data point, $\bar{x}$ refers to the mean of all data points, and n is the number of data points. In the case of the cv, it measures the ratio between the mean value and the variance of a data sample and can be defined as in 5.4. For the data sample p, which includes five pose estimations, the cv is calculated as:

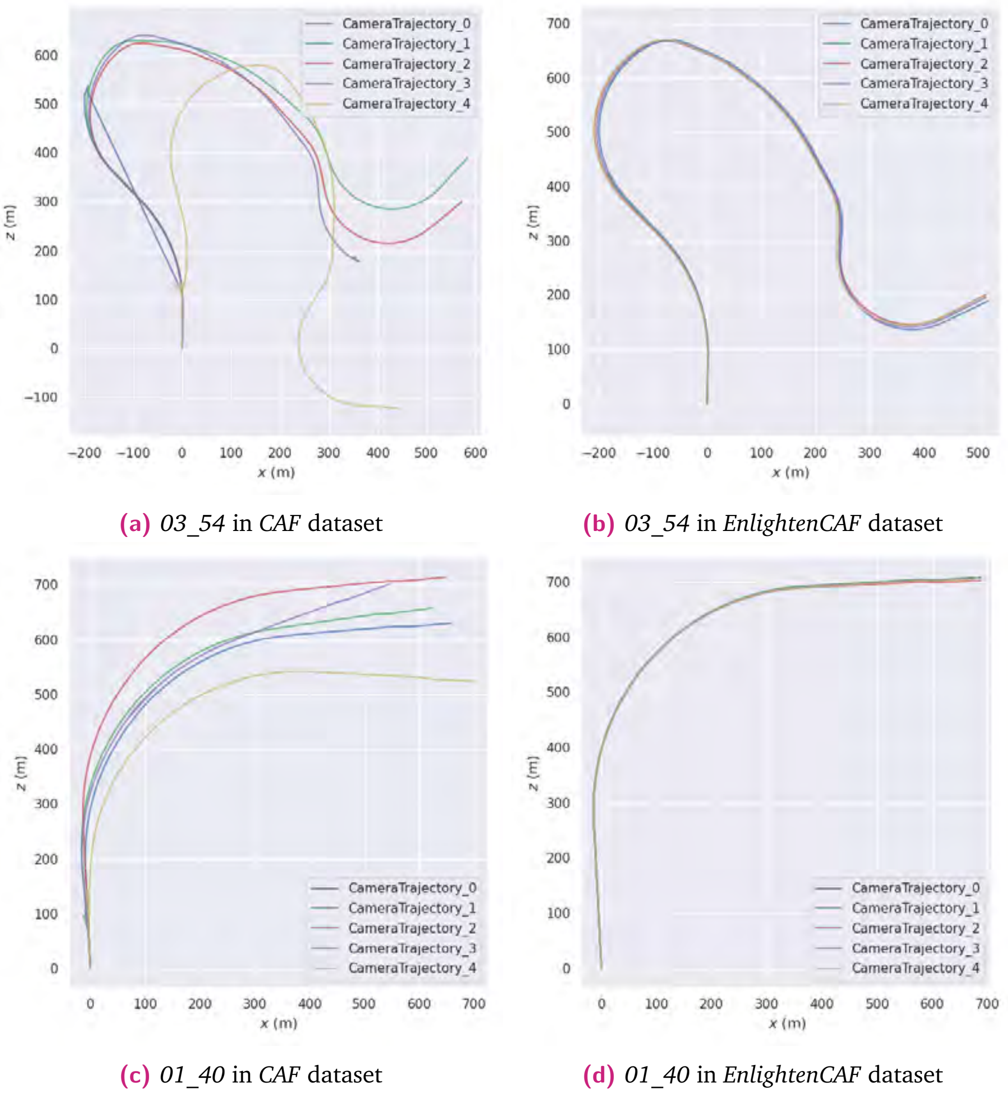
$$cv = \frac{\sigma^2}{\bar{x}}100 \tag{5.4}$$

where σ^2 and $\bar{x}$ are the variance and the mean of all data points, respectively. The lower the cv value, the lower the dispersion. A value of $cv < 1$ is considered to be a low variation sample.

The dispersion of the poses estimated by ORB-SLAM2 in different runs seems to reduce when enhancing the frames with EnlightenGAN. Therefore, it can be concluded that the translation estimation error range of the non-deterministic ORB-SLAM2 algorithm can be reduced by applying GAN based data enhancement techniques such as EnlightenGAN.

The evaluation has been done by executing ORB-SLAM2 in both datasets, the original *CAF* and the enhanced *EnlightenCAF*, to compare the dispersion of the estimated poses among different executions. Each sequence has been run five times as proposed by [91]. Therefore, the number of points for each pose when calculating the variance (σ^2) and the coefficient of variation (cv) is $n = 5$.

Figure 5.14 shows the results of applying ORB-SLAM2 five times on trajectory *01_54* in the *CAF* and the enhanced *EnlightenCAF* datasets. It can be seen that the distribution of the poses through the trajectory is more constant in the enlightened dataset.



(a) *03_54* in *CAF* dataset

(b) *03_54* in *EnlightenCAF* dataset

(c) *01_40* in *CAF* dataset

(d) *01_40* in *EnlightenCAF* dataset

Figure 5.14.: Pose dispersion analysis on sample sequences *01_40* and *03_54*. ORB-SLAM2 algorithm is executed five times on each dataset.

Regarding the ORB-SLAM2 dispersion, seems that the scenarios' lighting conditions are a critical characteristic for ORB-SLAM2 performance, and that improving that lighting through data enhancement methods reduces the pose estimation dispersion. Good lighting conditions seem to be critical for VO/vSLAM algorithms.

From the results, it can be seen that enlightening the datasets with *EnlightenGAN* increases the DF-VO and ORB-SLAM2 performance. Moreover, it tends to reduce ORB-SLAM2 dispersion in pose estimation.

5.6 Conclusion

This chapter covers the experimentation related to the DF-VO and ORB-SLAM2 performance in the lighting enhanced proprietary datasets *EnlightenCarDriving* and *EnlightenCAF*. The urban underground railway domain is characterized by varying lighting conditions (tunnel vs. platform), low illumination (in tunnels), or texture-less areas that challenged the state-of-the-art VO/vSLAM algorithms. These algorithms are mainly based on scenario feature extraction processes that depend on image luminosity. Therefore, their performance in the presented low-light scenarios diverges from the performance in standard VO/vSLAM benchmarks. However, new approaches from the data enhancement research community were identified as a methodology to reduce the impact of poor lighting conditions in VO/vSLAM algorithms.

The results show that the data enhancement through EnlightenGAN increases the performance of both state-of-the-art DF-VO and ORB-SLAM2 algorithms in all the generated enhanced datasets (*EnlightenCarDriving* and *EnlightenCAF*), reducing the Absolute Trajectory Error by at least 1.91% and the Relative Translational Error by 2.80% when using the enhanced frames.

Furthermore, as ORB-SLAM2 is considered non-deterministic due to its initialization phase, the pose estimations can vary from run to run. Analyzing different ORB-SLAM2 runs in the same sequences, it was detected that GAN-based enhanced techniques tend to reduce the dispersion of the pose estimations among different runs. A deeper analysis of the frames where the dispersion of ORB-SLAM2 increases or decreases could lead to detecting the scenario characteristics that hinder the use of such VO/vSLAM algorithms and, therefore, explore adjustments to increase their performance in such challenging domains.

6 Conclusions

This research aims to investigate the usage of Computer Vision (CV) for autonomous train operations in the urban underground railway domain. An autonomous train must accomplish all the operations in an autonomous way including the localization. In that task, the use of cameras could be helpful as it has been in other domains such as robotics or other vehicles (e.g., cars or drones).

The research in the railway domain has recently raised due to the transformation of railway vehicles toward autonomous driving systems. However, it is a starting research field, with little specific work in train localization and no focus on the cameras as the primary sensors. Most of works have focused on sensor fusion [49]–[54]. To our knowledge, this research is the first research work that explores the application of VO/vSLAM algorithms based only on cameras as primary sensors in the underground railway scenarios. This decision comes from the relative low-cost of cameras and the results shown in other domains. The urban underground railway domain is characterized by varying lighting conditions (tunnel vs. platform), low illumination (in tunnels), or texture-less areas that challenge the VO/vSLAM in this domain.

When focusing on the literature in the railway domain, limited research works have been found. Therefore, a state of the art (SOTA) of CV-based autonomous robot and vehicle localization algorithms has been done, pursuing the usage of state-of-the-art algorithms in the target domain. The SOTA points out that the localization algorithms started relying on the geometric features of the images in the early 2000s, and then, learning-based algorithms proliferated due to the Machine Learning advances.

Learning-based algorithms have mainly focused on Convolutional Neural Networks (CNN) for their feature-extraction capabilities and Recurrent Neural Networks (RNN) to include temporal knowledge through sequential modeling. The challenges faced by some geometric VO/vSLAM algorithms, such as the scale inconsistency or the drift issues of monocular algorithms, have been focused on by learning the depth and optical flow from image sequences. Furthermore, as most recent learning VO/vSLAM algorithms are based on learning scene depth and flow between consecutive frames, it has become a research direction itself. However, these algorithms have to deal with other problems such as the perceptually tough scenario characteristics or the adaptability to unknown environments.

The literature review of VO/vSLAM in visually degraded environments and scenarios evidences the need for further research in this direction. State-of-the-art algorithms have been primarily tested in standard scenarios, but their performance is not measured in scenarios with different circumstances (e.g., poor lighting conditions, non-Lambertian surfaces, textureless areas or dynamic environments). Consequently, and based on this analysis, this research has focused on the study of the performance of VO/vSLAM algorithms from robotics and autonomous vehicles in a visually challenging scenario: the urban underground railway domain.

6.1 General conclusions

Reviewing the most referenced datasets by the VO/vSLAM research community, no dataset was found to fit the target scenario. Most of the referenced datasets belong to the robotics or car domains. Moreover, most state-of-the-art VO/vSLAM algorithms have been evaluated on the KITTI vision benchmark [11] which offers data captured from a moving car in outdoor urban scenarios. These scenarios are defined in outdoor environments with good lighting settings and favorable conditions for feature extraction.

Among the other analyzed datasets, it should be noted that only one database related to autonomous localization and mapping covers the railway domain (Nordland [190]), although it covers only outdoor scenarios. Consequently, the need for a proprietary dataset from the target domain arose. The *CAF* dataset was generated by recording in an urban underground railway scenario.

Furthermore, given the access difficulties to the target scenario, the complementary dataset *CarDriving* has been recorded to validate the recording setup (e.g., camera setup, frame quality,...). This dataset was generated from a driving car in an urban environment recording by night, imitating the low-light conditions of the urban underground railway scenario.

Usually, the ground truth for standard datasets is generated by adding a GPS-based localization system to the cameras. The ground truth for the *CarDriving* dataset has been obtained from the standard algorithm ORB-SLAM2 and corrected with GPS data. However, in the railway scenario, the GPS is unavailable and, therefore, another methodology to acquire reference data was pursued. Ground truth has been generated by designing and implementing a novel method to estimate 6-DoF poses for the frames recorded in an underground railway line. The generation process is based on the synchronization of geodetic coordinates from a geomap, ATP data recorded from the train wheel sensors, and a railway gradient map provided by the railway constructor. This ground truth generation method shows that reference data can be generated from other sources than the most commonly used GPS.

To our knowledge, the *CAF* dataset is the first VO/vSLAM dataset in the urban underground railway domain. In order to make it scientifically sound, it follows the standard KITTI format and data volume. The generated *CAF* dataset enables the evaluation of the state-of-the-art VO/vSLAM algorithms in scenarios with challenging perceptual characteristics that can not be found in the standard VO/vSLAM datasets.

From the SOTA of VO/vSLAM algorithms, DF-VO [133] and ORB-SLAM2 [91] were selected for the experimentation as they were two of the algorithms with the best results in robotics and autonomous car datasets such as KITTI, and they belong to distinct types of VO/vSLAM algorithms (learning-based and geometric, respectively).

The evaluation of state-of-the-art VO/vSLAM algorithms in the generated datasets *CAF* and *CarDriving* has been conducted following standard VO/vSLAM evaluation metrics. It can be seen that the results in the *CarDriving* dataset are similar to the results obtained in the standard KITTI dataset.

However, the performance of state-of-the-art DF-VO and ORB-SLAM2 algorithms is considerably degraded in the *CAF* dataset. Furthermore, by shortening the underground railway sequences to platform areas with more similar lighting conditions to standard datasets such as KITTI, the errors are also reduced to similar values of executing the VO/vSLAM algorithms in those standard datasets.

As mentioned in [211], geometry-based VO algorithms such as ORB-SLAM2 suffer a scale drift issue when ideal visual conditions are not met. The results obtained in the *CAF* dataset raise the need for affording scale drift and low lighting-related visual conditions. They require an adaptation procedure of this type of VO/vSLAM algorithm to handle scenarios that contain more challenging visual characteristics (low-light, low-textures, or non-Lambertian surfaces).

In the case of DF-VO, being a hybrid algorithm, the scale may be wrongly estimated due to issues related to the geometric characteristics of the underground visual domain or the deep learning training process. Even training the depth estimation network, the scenario lighting characteristics (low-light) and the repetitive textures seem to hinder the performance of the depth estimation algorithm compared to standard datasets that do not afford such challenging visual characteristics.

Nevertheless, these results obtained by the reference algorithms in the *CAF* dataset demonstrate that an adaptation of reference VO solutions is required in order to become applicable in challenging scenarios, such as the underground railway domain. Following the above idea, this research also explores improving those results in the target domain through image enhancement techniques. New approaches from data enhancement research have brought the opportunity to analyze if they might reduce the impact of poor lighting conditions in VO/vSLAM algorithms.

Therefore, enhanced dataset variants of *CAF* (*EnlightenCAF*) and *CarDriving* (*EnlightenCarDriving*) datasets have been generated using EnlightenGAN [242] algorithm.

The results of using DF-VO and ORB-SLAM2 in the enlightened proprietary datasets (*EnlightenCarDriving* and *EnlightenCAF*) show that the data enhancement through EnlightenGAN increases the pose estimation accuracy, reducing the translational error by at least 18% compared to the non-enhanced datasets.

Furthermore, from a first evaluation of the pose estimations variability of the non-deterministic ORB-SLAM2 algorithm, it can be seen that enlightening the sequences tend to reduce the dispersion of the ORB-SLAM2 estimations.

6.2 Future work

This research evidences the need of further research activities in challenging scenarios such as in the underground railway domain. It covers the analysis of two state-of-the-art algorithms (DF-VO and ORB-SLAM2), but the exploration of other algorithms is foreseen as a future research task. An option is the exploration of geometric direct VO/vSLAM algorithms in such challenging domains (e.g., Stereo DSO [82] (2017) or TANDEM [83] (2022)), which has not been analyzed in this research. Other options include the use of VO/vSLAM works that have been designed to handle other challenging characteristics as [89] that focuses on scenarios under day-night or seasonal changes; in [131] BGNet is proposed, an algorithm that handles scenes with illumination changes or repetitive patterns; [121] proposes GeoNet, a robust algorithm for non-Lambertian surfaces; or, low-textured regions are explored in CNN-SLAM [102].

Furthermore, lately, research works in the VO/vSLAM community have pointed out data fusion as the best solution to autonomous vehicles' localization [15], [16].

Therefore, the inclusion of other sensors, such as the LiDAR or an IMU, is also foreseen.

Focusing the proposed ground truth generation method, an interesting approach is to improve the train stopping point detection in the image sequences by updating the SSIM with other methods (i.e., the Siamese networks) that could lead to a more automatic ground truth generation method skipping the SSIM threshold definition. Also, the adaptation of this reference data generation method to other underground scenarios might be interesting to evaluate the suitability of the proposed dataset generation method.

Referring to the results obtained by the selected state-of-the-art VO/vSLAM algorithms, the estimation error of the learning part of DF-VO could be reduced by training the deep models in the target scenario. However, the authors mention that the depth network under DF-VO (Monodepth2) may not be appropriate for these challenging scenarios as it might be difficult for self-supervised algorithms to learn in scenarios with visually degraded conditions. This evidences a research direction to explore learning depth in such scenarios where the lighting conditions are inadequate, with textureless areas or non-Lambertian surfaces.

From the results obtained by DF-VO and ORB-SLAM2 in the enhanced datasets, the exploration of other enhancing algorithms can be an interesting research field. Several research works are envisioned, such as the application of more types of enhancement methods (e.g., algorithms devoted to other visually challenging issues such as the non-Lambertian surfaces), the integration of the enhancement with the VO/vSLAM algorithms as a unique process or the research works focused on learning the best enhancement combination for the target scenario.

Furthermore, a deeper analysis of the frames where the dispersion of ORB-SLAM2 increases or decreases could lead to detecting the scenario characteristics that hinder the use of such VO/vSLAM algorithms and, therefore, explore adjustments to increase their performance in such challenging domains.

Lastly, the analysis of the application of VO/vSLAM in the embedded systems of the train is out of the scope of this research. Evaluating the performance of the algorithms in such systems is an engaging research direction. Furthermore, current train localization systems are designed according to functional safety standards, that currently do not consider the application of deep learning-based algorithms.

References

[1] R. Klette, *Concise computer vision*. Springer, 2014 (cit. on p. 1).

[2] M. Irani, B. Rousso, and S. Peleg, *Recovery of ego-motion using image stabilization*. Hebrew University of Jerusalem. Leibniz Center for Research in Computer ..., 1993 (cit. on p. 1).

[3] G. P. Stein, O. Mano, and A. Shashua, "A robust method for computing vehicle ego-motion," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511)*, IEEE, 2000, pp. 362–368 (cit. on p. 1).

[4] K. Yamaguchi, T. Kato, and Y. Ninomiya, "Vehicle ego-motion estimation and moving object detection using a monocular camera," in *18th International Conference on Pattern Recognition (ICPR'06)*, IEEE, vol. 4, 2006, pp. 610–613 (cit. on p. 1).

[5] F. Tschopp, T. Schneider, A. W. Palmer, *et al.*, "Experimental Comparison of Visual-Aided Odometry Methods for Rail Vehicles," *IEEE Robotics and Automation Letters*, pp. 1–1, 2019. arXiv: `arXiv:1904.00936v1` (cit. on pp. 1, 7, 15, 27).

[6] V. Grabe, H. H. Bülthoff, D. Scaramuzza, and P. R. Giordano, "Nonlinear ego-motion estimation from optical flow for online control of a quadrotor uav," *The International Journal of Robotics Research*, vol. 34, no. 8, pp. 1114–1135, 2015 (cit. on p. 1).

[7] T. Albrecht, K. Lüddecke, and J. Zimmermann, "A precise and reliable train positioning system and its use for automation of train operation," *IEEE ICIRT 2013 - Proceedings: IEEE International Conference on Intelligent Rail Transportation*, pp. 134–139, 2013 (cit. on p. 1).

[8] "Ieee standard for communications-based train control (cbtc) performance and functional requirements," *IEEE Std 1474.1-2004 (Revision of IEEE Std 1474.1-1999)*, pp. 1–45, 2004 (cit. on p. 1).

[9] M. Malvezzi, B. Allotta, and M. Rinchi, "Odometric estimation for automatic train protection and control systems," *Vehicle System Dynamics*, vol. 49, no. 5, pp. 723–739, 2011 (cit. on p. 2).

[10] J. Marais, J. Beugin, and M. Berbineau, "A Survey of GNSS-Based Research and Developments for the European Railway Signalling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2602–2618, 2017 (cit. on p. 2).

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013 (cit. on pp. 2, 3, 24, 29, 30, 38, 41, 90).

[12] M. Burri, J. Nikolic, P. Gohl, *et al.*, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016 (cit. on pp. 2, 30).

[13] C. G. Atkeson, P. B. Benzun, N. Banerjee, *et al.*, "Achieving reliable humanoid robot operations in the darpa robotics challenge: Team wpi-cmu's approach," in *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*, Springer, 2018, pp. 271–307 (cit. on pp. 2, 25).

[14] T. Rouček, M. Pecka, P. Čížek, *et al.*, "Darpa subterranean challenge: Multi-robotic exploration of underground environments," in *International Conference on Modelling and Simulation for Autonomous Systesm*, Springer, Cham, 2019, pp. 274–290 (cit. on pp. 2, 25).

[15] K. Ebadi, Y. Chang, M. Palieri, *et al.*, "Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 80–86 (cit. on pp. 2, 25, 26, 92).

[16] A. Agha, K. Otsu, B. Morrell, *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021 (cit. on pp. 2, 25, 26, 92).

[17] J.-L. Blanco, F.-A. Moreno, and J. Gonzalez-Jimenez, "The málaga urban dataset: High-rate stereo and lidars in a realistic urban scenario," *International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2014 (cit. on pp. 3, 30, 41).

[18] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017 (cit. on pp. 3, 30, 41).

[19] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011 (cit. on pp. 3, 30, 41).

[20] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019 (cit. on pp. 3, 30, 41).

[21] G. Giralt, R. Chatila, and M. Vaisset, "An integrated navigation and motion control system for autonomous multisensory mobile robots," in *Autonomous robot vehicles*, Springer, 1990, pp. 420–443 (cit. on p. 5).

[22] W. Bernhart and M. Winterhoff, "Autonomous driving: Disruptive innovation that promises to change the automotive industry as we know it," in *Energy Consumption and Autonomous Driving*, Springer, 2016, pp. 3–10 (cit. on p. 5).

[23] P. Goel, S. I. Roumeliotis, and G. S. Sukhatme, "Robust localization using relative and absolute position estimates," in *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No. 99CH36289)*, IEEE, vol. 2, 1999, pp. 1134–1140 (cit. on p. 5).

[24] A. Yol, B. Delabarre, A. Dame, J.-E. Dartois, and E. Marchand, "Vision-based absolute localization for unmanned aerial vehicles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 3429–3434 (cit. on p. 5).

[25] B.-K. Cho and S.-H. Ryu, "Performance evaluation of video based obstacle detection algorithm for railway level crossing," in *Future Information Technology*, Springer, 2014, pp. 909–914 (cit. on p. 6).

[26] M. Karakose, O. Yaman, M. Baygin, K. Murat, and E. Akin, "A new computer vision based method for rail track detection and fault diagnosis in railways," *International Journal of Mechanical Engineering and Robotics Research*, vol. 6, no. 1, pp. 22–17, 2017 (cit. on p. 6).

[27] S. Mittal and D. Rao, "Vision based railway track monitoring using deep learning," *arXiv preprint arXiv:1711.06423*, 2017 (cit. on p. 6).

[28] M. Wedberg, *Detecting rails in images from a train-mounted thermal camera using a convolutional neural network*, 2017 (cit. on p. 6).

[29] S. Sawadisavi, J. R. Edwards, E. Resendiz, *et al.*, "Machine-vision inspection of railroad track," in *Proceedings of the TRB 88th Annual Meeting, Washington, DC*, 2009, pp. 1–19 (cit. on p. 6).

[30] X. Gibert, V. M. Patel, and R. Chellappa, "Deep multitask learning for railway track inspection," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 1, pp. 153–164, 2016 (cit. on p. 6).

[31] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, and B. De Schutter, "Deep convolutional neural networks for detection of rail surface defects," in *2016 International joint conference on neural networks (IJCNN)*, IEEE, 2016, pp. 2584–2589 (cit. on p. 6).

[32] A. K. Singh, A. Swarup, A. Agarwal, and D. Singh, "Vision based rail track extraction and monitoring through drone imagery," *Ict Express*, vol. 5, no. 4, pp. 250–255, 2019 (cit. on p. 6).

[33] B. T. Nassu and M. Ukai, "Rail extraction for driver support in railways," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, pp. 83–88 (cit. on p. 6).

[34] F. Maire and A. Bigdeli, "Obstacle-free range determination for rail track maintenance vehicles," in *2010 11th International Conference on Control Automation Robotics & Vision*, IEEE, 2010, pp. 2172–2178 (cit. on p. 6).

[35] L. F. Rodriguez, J. A. Uribe, and J. V. Bonilla, "Obstacle detection over rails using hough transform," in *2012 XVII Symposium of Image, Signal Processing, and Artificial Vision (STSIVA)*, IEEE, 2012, pp. 317–322 (cit. on p. 6).

[36] A. Berg, K. Öfjäll, J. Ahlberg, and M. Felsberg, "Detecting rails and obstacles using a train-mounted thermal camera," in *Scandinavian Conference on Image Analysis*, Springer, 2015, pp. 492–503 (cit. on p. 6).

[37] M. Etxeberria-Garcia, M. Labayen, M. Zamalloa, and N. Arana-Arexolaleiba, “Application of computer vision and deep learning in the railway domain for autonomous train stop operation,” in *2020 IEEE/SICE International Symposium on System Integration (SII)*, IEEE, 2020, pp. 943–948 (cit. on p. 6).

[38] M. Etxeberria-Garcia, M. Labayen, F. Eizaguirre, M. Zamalloa, and N. Arana-Arexolaleiba, “Monocular visual odometry for underground railway scenarios,” in *Fifteenth International Conference on Quality Control by Artificial Vision*, International Society for Optics and Photonics, vol. 11794, 2021, p. 1 179 402 (cit. on p. 6).

[39] J. P. N. Acilo, A. G. S. D. Cruz, M. K. L. Kaw, *et al.*, “Traffic sign integrity analysis using deep learning,” in *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, IEEE, 2018, pp. 107–112 (cit. on p. 6).

[40] A. D. Kumar, “Novel deep learning model for traffic sign detection using capsule networks,” *arXiv preprint arXiv:1805.04424*, 2018 (cit. on p. 6).

[41] G. Karagiannis, S. Olsen, and K. Pedersen, “Deep learning for detection of railway signs and signals,” in *Science and Information Conference*, Springer, 2019, pp. 1–15 (cit. on p. 6).

[42] M. Etxeberria-Garcia, F. Ezaguirre, J. Plazaola, U. Munoz, and M. Zamalloa, “Embedded object detection applying deep neural networks in railway domain,” in *2020 23rd Euromicro Conference on Digital System Design (DSD)*, IEEE, 2020, pp. 565–569 (cit. on p. 6).

[43] M. A. Haseeb, J. Guan, D. Ristić-Durrant, and A. Gräser, “Disnet: A novel method for distance estimation from monocular camera,” 2012 (cit. on p. 6).

[44] H. M. Abdul, R.-D. Danijela, G. Axel, B. Milan, and S. Dušan, “Multi-disnet: Machine learning-based object distance estimation from multiple cameras,” in *International Conference on Computer Vision Systems*, Springer, 2019, pp. 457–469 (cit. on p. 6).

[45] H. Alawad, S. Kaewunruen, and M. An, “A deep learning approach towards railway safety risk assessment,” *IEEE Access*, vol. 8, pp. 102 811–102 832, 2020 (cit. on p. 6).

[46] E. Páli, K. Mathe, L. Tamas, and L. Buşoniu, “Railway track following with the ar. drone using vanishing point detection,” in *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, IEEE, 2014, pp. 1–6 (cit. on p. 6).

[47] X. Jiang, X. Luo, S. Luo, G. Lyu, and S. Wang, “A straight-line-based vanishing point detection method for railway environmental images,” in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, IEEE, 2016, pp. 737–741 (cit. on p. 6).

[48] S. Oh, S. Park, and C. Lee, “Vision based platform monitoring system for railway station safety,” in *2007 7th International Conference on ITS Telecommunications*, IEEE, 2007, pp. 1–5 (cit. on p. 7).

[49] D. Burschka, C. Robl, and S. Ohrendorf-Weiss, “Optical navigation in unstructured dynamic railroad environments,” *arXiv preprint arXiv:2007.03409*, 2020 (cit. on pp. 7, 27, 89).

[50] F. Tschopp, C. von Einem, A. Cramariuc, *et al.*, “Hough^{2}map–iterative event-based hough transform for high-speed railway mapping,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2745–2752, 2021 (cit. on pp. 7, 27, 89).

[51] Y. LOU, Y. WANG, Z. TU, Y. ZHANG, and W. SONG, “Real time localization and mapping integrating multiple prism lidars/imu/rtk on railway locomotive,” vol. 46, no. 12, pp. 1802–1807, 2021 (cit. on pp. 7, 27, 89).

[52] Y. Wang, W. Song, Y. Lou, *et al.*, “Simultaneous location of rail vehicles and mapping of environment with multiple lidars,” *arXiv preprint arXiv:2112.13224*, 2021 (cit. on pp. 7, 27, 89).

[53] Y. Wang, Y. Lou, Y. Zhang, *et al.*, “Raillomer: Rail vehicle localization and mapping with lidar-imu-odometer-gnss data fusion,” *arXiv preprint arXiv:2111.15043*, 2021 (cit. on pp. 7, 27, 89).

[54] J. Liu, X.-L. Zhao, B.-G. Cai, and J. Wang, “Pseudolite constellation optimization for seamless train positioning in gnss-challenged railway stations,” *IEEE Transactions on Intelligent Transportation Systems*, 2021 (cit. on pp. 7, 27, 89).

[55] F. Fraundorfer and D. Scaramuzza, “Visual odometry: Part i: The first 30 years and fundamentals,” *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011 (cit. on pp. 7, 8).

[56] R. C. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986 (cit. on p. 7).

[57] T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix, “Vision-based slam: Stereo and monocular approaches,” *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, 2007 (cit. on p. 7).

[58] X. Gao, T. Zhang, Y. Liu, and Q. Yan, *14 lectures on visual slam: From theory to practice*, 2017 (cit. on p. 7).

[59] R. Gonzalez, F. Rodriguez, J. L. Guzman, C. Pradalier, and R. Siegwart, “Combined visual odometry and visual compass for off-road mobile robots localization,” *Robotica*, vol. 30, no. 6, pp. 865–878, 2012 (cit. on pp. 8, 26).

[60] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, “Review of visual odometry: Types, approaches, challenges, and applications,” *SpringerPlus*, vol. 5, no. 1, pp. 1–26, 2016 (cit. on pp. 8, 24).

[61] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946 (cit. on pp. 8, 10, 18, 20, 30).

[62] A. Kendall and R. Cipolla, “Geometric loss functions for camera pose regression with deep learning,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6555–6564, 2017 (cit. on pp. 8, 10, 16, 18, 20).

[63] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, “An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics,” *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015 (cit. on p. 8).

[64] D. Nistér, O. Naroditsky, and J. Bergen, “Visual Odometry,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004, p. 1 (cit. on p. 8).

[65] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981 (cit. on pp. 8, 15).

[66] M. Pollefeys, D. Nistér, J.-M. Frahm, *et al.*, “Detailed real-time urban 3d reconstruction from video,” *International Journal of Computer Vision*, vol. 78, no. 2-3, pp. 143–167, 2008 (cit. on p. 8).

[67] K. Konolige, M. Agrawal, and J. Sola, “Large-scale visual odometry for rough terrain,” in *Robotics research*, Springer, 2010, pp. 201–212 (cit. on p. 8).

[68] J.-P. Tardif, M. George, M. Laverne, A. Kelly, and A. Stentz, “A new approach to vision-aided inertial navigation,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 4161–4168 (cit. on p. 8).

[69] E. S. Jones and S. Soatto, “Visual-inertial navigation, mapping and localization: A scalable real-time causal approach,” *The International Journal of Robotics Research*, vol. 30, no. 4, pp. 407–430, 2011 (cit. on p. 8).

[70] J.-C. Piao and S.-D. Kim, “Real-time visual–inertial slam based on adaptive keyframe selection for mobile ar applications,” *IEEE Transactions on Multimedia*, vol. 21, no. 11, pp. 2827–2836, 2019 (cit. on p. 8).

[71] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, “D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry,” *arXiv preprint arXiv:2003.01060*, 2020 (cit. on pp. 9, 10, 19, 23, 24, 29, 40).

[72] Y. Wang and Y.-F. Xu, “Unsupervised learning of accurate camera pose and depth from video sequences with kalman filter,” *Ieee Access*, vol. 7, pp. 32 796–32 804, 2019 (cit. on pp. 9, 10, 22, 23).

[73] J. Zhang, Q. Su, P. Liu, C. Xu, and Y. Chen, “Unsupervised learning of monocular depth and ego-motion with space–temporal-centroid loss,” *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 3, pp. 615–627, 2020 (cit. on pp. 9, 10, 22, 23).

[74] Y. Zou, P. Ji, Q.-H. Tran, J.-B. Huang, and M. Chandraker, “Learning monocular visual odometry via self-supervised long-term modeling,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, Springer, 2020, pp. 710–727 (cit. on p. 9).

[75] K. Wang, S. Ma, J. Chen, F. Ren, and J. Lu, “Approaches challenges and applications for deep visual odometry toward to complicated and emerging areas,” *IEEE Transactions on Cognitive and Developmental Systems*, 2020 (cit. on p. 9).

[76] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, “Dtam: Dense tracking and mapping in real-time,” in *2011 international conference on computer vision*, IEEE, 2011, pp. 2320–2327 (cit. on pp. 10, 12, 14).

[77] J. Engel, J. Sturm, and D. Cremers, “Semi-dense visual odometry for a monocular camera,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1449–1456, 2013 (cit. on pp. 10, 12, 14).

[78] J. Engel, T. Schops, and D. Cremers, “LSD-SLAM: Large-Scale Direct Monocular SLAM,” in *European conference on computer vision*, Springer, Cham, 2014, pp. 834–849 (cit. on pp. 10, 12, 14).

[79] C. Forster, M. Pizzoli, and D. Scaramuzza, “Svo: Fast semi-direct monocular visual odometry,” in *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2014, pp. 15–22 (cit. on pp. 10–12, 14, 19).

[80] H. Alismail, M. Kaess, B. Browning, and S. Lucey, “Direct visual odometry in low light using binary descriptors,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 444–451, 2016 (cit. on pp. 10, 12, 14).

[81] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017 (cit. on pp. 10, 12, 14, 63).

[82] R. Wang, M. Schworer, and D. Cremers, “Stereo dso: Large-scale direct sparse visual odometry with stereo cameras,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3903–3911 (cit. on pp. 10, 12, 14, 92).

[83] L. Koestler, N. Yang, N. Zeller, and D. Cremers, “Tandem: Tracking and dense mapping in real-time using deep multi-view stereo,” in *Conference on Robot Learning*, PMLR, 2022, pp. 34–45 (cit. on pp. 10, 12, 14, 92).

[84] S. Se, D. Lowe, and J. Little, “Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks,” *The international Journal of robotics Research*, vol. 21, no. 8, pp. 735–758, 2002 (cit. on pp. 10–12).

[85] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007 (cit. on pp. 10–12, 63).

[86] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, IEEE, 2007, pp. 225–234 (cit. on pp. 10, 12, 13, 67).

[87] A. Geiger, J. Ziegler, and C. Stiller, “Stereoscan: Dense 3d reconstruction in real-time,” in *2011 IEEE intelligent vehicles symposium (IV)*, Ieee, 2011, pp. 963–968 (cit. on pp. 10, 12, 13).

[88] H. Badino, A. Yamamoto, and T. Kanade, “Visual odometry by multi-frame feature integration,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 222–229 (cit. on pp. 10, 12, 13).

[89] E. Stenborg, T. Sattler, and L. Hammarstrand, “Using image sequences for long-term visual localization,” in *2020 International Conference on 3D Vision (3DV)*, IEEE, 2020, pp. 938–948 (cit. on pp. 10, 12, 25, 92).

[90] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, “ORB-SLAM: A Versatile and Accurate Monocular SLAM System,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015. arXiv: `arXiv:1502.00956v2` (cit. on pp. 10, 12, 13).

[91] R. Mur-Artal and J. D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017 (cit. on pp. 10, 12, 13, 27, 55, 60, 61, 63, 73, 86, 91).

[92] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, “Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam,” *IEEE Transactions on Robotics*, 2021 (cit. on pp. 10, 12, 13).

[93] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. Balch, “Memory-based learning for visual odometry,” in *2008 IEEE International Conference on Robotics and Automation*, IEEE, 2008, pp. 47–52 (cit. on pp. 10, 15).

[94] V. Guizilini and F. Ramos, “Semi-parametric learning for visual odometry,” *The International Journal of Robotics Research*, vol. 32, no. 5, pp. 526–546, 2013 (cit. on pp. 10, 15).

[95] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, “Evaluation of non-geometric methods for visual odometry,” *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1717–1730, 2014 (cit. on pp. 10, 15, 53).

[96] E. Brachmann, A. Krull, S. Nowozin, *et al.*, “Dsac-differentiable ransac for camera localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6684–6692 (cit. on pp. 10, 15).

[97] D. Barath and J. Matas, “Graph-cut ransac,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6733–6741 (cit. on pp. 10, 15).

[98] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, “Exploring representation learning with cnns for frame-to-frame ego-motion estimation,” *IEEE robotics and automation letters*, vol. 1, no. 1, pp. 18–25, 2015 (cit. on pp. 10, 16, 20).

[99] B. Ummenhofer, H. Zhou, J. Uhrig, *et al.*, “DeMoN: Depth and Motion Network for Learning Monocular Stereo,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 50.8–5047 (cit. on pp. 10, 19, 20).

[100] S. Wang, R. Clark, H. Wen, and N. Trigoni, “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 2043–2050 (cit. on pp. 10, 18, 20, 24, 29).

[101] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, “VidLoc : A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6856–6864 (cit. on pp. 10, 18, 20).

[102] K. Tateno, F. Tombari, I. Laina, and N. Navab, “CNN-SLAM : Real-time dense monocular SLAM with learned depth prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6243–6252 (cit. on pp. 10, 17, 19, 20, 25, 92).

[103] F. Walch, C. H. L. Leal-taix, T. Sattler, and S. H. D. Cremers, “Image-based localization using LSTMs for structured feature correlation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 627–637 (cit. on pp. 10, 20, 30).

[104] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Toward geometric deep slam,” *arXiv preprint arXiv:1707.07410*, 2017 (cit. on pp. 10, 16, 20).

[105] V. Peretroukhin, L. Clement, and J. Kelly, “Inferring sun direction to improve visual odometry: A deep learning approach,” *The International Journal of Robotics Research*, vol. 37, no. 9, pp. 996–1016, 2018 (cit. on pp. 10, 19, 20).

[106] G. Costante and T. A. Ciarfuglia, “Ls-vo: Learning dense optical subspace for robust visual odometry estimation,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1735–1742, 2018 (cit. on pp. 10, 17, 19, 20, 40, 73).

[107] J. Jiao, J. Jiao, Y. Mo, W. Liu, and Z. Deng, “Magicvo: End-to-end monocular visual odometry through deep bi-directional recurrent convolutional neural network,” *arXiv preprint arXiv:1811.10964*, 2018 (cit. on pp. 10, 18, 20).

[108] S. Wang, R. Clark, H. Wen, and N. Trigoni, “End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks,” *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018 (cit. on pp. 10, 19, 20).

[109] A. Valada, N. Radwan, and W. Burgard, “Deep auxiliary learning for visual localization and odometry,” in *International Conference on Robotics and Automation (ICRA 2018)*, IEEE, 2018 (cit. on pp. 10, 18, 20).

[110] Y. Lin, Z. Liu, J. Huang, *et al.*, “Deep global-relative networks for end-to-end 6-dof visual localization and odometry,” in *Pacific Rim International Conference on Artificial Intelligence*, Springer, 2019, pp. 454–467 (cit. on pp. 10, 19, 20).

[111] S. Y. Loo, A. J. Amiri, S. Mashohor, S. H. Tang, and H. Zhang, “Cnn-svo: Improving the mapping in semi-direct visual odometry using single-image depth prediction,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 5218–5223 (cit. on pp. 10, 17, 19, 20).

[112] G. Zhai, L. Liu, L. Zhang, Y. Liu, and Y. Jiang, “Poseconvgru: A monocular approach for visual ego-motion estimation by learning,” *Pattern Recognition*, vol. 102, p. 107 187, 2020 (cit. on pp. 10, 19, 20).

[113] L. Von Stumberg, P. Wenzel, N. Yang, and D. Cremers, “Lm-reloc: Levenberg-marquardt based direct visual relocalization,” in *2020 International Conference on 3D Vision (3DV)*, IEEE, 2020, pp. 968–977 (cit. on pp. 10, 17, 20).

[114] P.-E. Sarlin, A. Unagar, M. Larsson, *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3247–3257 (cit. on pp. 10, 19, 20).

[115] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017 (cit. on pp. 10, 20, 23).

[116] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1851–1858 (cit. on pp. 10, 23, 24, 29).

[117] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 7286–7291 (cit. on pp. 10, 18, 23).

[118] V. M. Babu, K. Das, A. Majumdar, and S. Kumar, "Undemon: Unsupervised deep network for depth and ego-motion estimation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1082–1088 (cit. on pp. 10, 21, 23).

[119] H. Zhan, R. Garg, C. Saroj Weerasekera, *et al.*, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 340–349 (cit. on pp. 10, 21, 23, 24, 29).

[120] A. Kathpal, D. Shah, and M. Pathak, "Learning the structure from motion, an unsupervised approach," 2018 (cit. on pp. 10, 21, 23).

[121] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1983–1992 (cit. on pp. 10, 17, 21, 23, 25, 92).

[122] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5667–5675 (cit. on pp. 10, 21, 23).

[123] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Self-improving visual odometry," *arXiv preprint arXiv:1812.03245*, 2018 (cit. on pp. 10, 17, 23).

[124] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 817–833 (cit. on pp. 10, 22–24, 29).

[125] V. Prasad and B. Bhowmick, "Sfmlearner++: Learning monocular depth & ego-motion using meaningful geometric constraints," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 2087–2096 (cit. on pp. 10, 21, 23).

[126] Q. Liu, R. Li, H. Hu, and D. Gu, “Using unsupervised deep learning technique for monocular visual odometry,” *Ieee Access*, vol. 7, pp. 18 076–18 088, 2019 (cit. on pp. 10, 21, 23).

[127] J. Bian, Z. Li, N. Wang, *et al.*, “Unsupervised scale-consistent depth and ego-motion learning from monocular video,” in *Advances in Neural Information Processing Systems*, 2019, pp. 35–45 (cit. on pp. 10, 21, 23).

[128] T. Shen, Z. Luo, L. Zhou, *et al.*, “Beyond photometric loss for self-supervised ego-motion estimation,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 6359–6365 (cit. on pp. 10, 21, 23).

[129] W. Zhao, S. Liu, Y. Shu, and Y.-J. Liu, “Towards better generalization: Joint depth-pose learning without posenet,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9151–9161 (cit. on pp. 10, 22, 23).

[130] S. Li, X. Wang, Y. Cao, *et al.*, “Self-supervised deep visual odometry with online adaptation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6339–6348 (cit. on pp. 10, 21, 23).

[131] H. Yu, W. Ye, Y. Feng, H. Bao, and G. Zhang, “Learning bipartite graph matching for robust visual localization,” in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2020, pp. 146–155 (cit. on pp. 10, 17, 23, 25, 92).

[132] S. Dong, Q. Fan, H. Wang, *et al.*, “Robust neural routing through space partitions for camera relocalization in dynamic indoor environments,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8544–8554 (cit. on pp. 10, 22, 23).

[133] H. Zhan, C. S. Weerasekera, J.-W. Bian, R. Garg, and I. Reid, “Df-vo: What should be learnt for visual odometry?” *arXiv preprint arXiv:2103.00933*, 2021 (cit. on pp. 10, 17, 22–24, 27, 29, 55, 56, 60, 61, 63, 73, 91).

[134] C. G. Harris, M. Stephens, *et al.*, “A combined corner and edge detector.,” in *Alvey vision conference*, Citeseer, vol. 15, 1988, pp. 10–5244 (cit. on p. 11).

[135] B. Everett and L. Feng, “Navigating mobile robots: Systems and techniques,” *AK Peters, Ltd. Natick, MA, USA*, 1996 (cit. on p. 11).

[136] H. Gaoussou and P. Dewei, “Evaluation of the Visual Odometry Methods for Semi-Dense Real-Time,” *Advanced Computing: An International Journal*, vol. 9, no. 2, pp. 01–14, 2018 (cit. on p. 11).

[137] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004 (cit. on p. 11).

[138] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *European conference on computer vision*, Springer, 2006, pp. 430–443 (cit. on p. 11).

[139] A. Vakhitov, V. Lempitsky, and Y. Zheng, “Stereo relative pose from line and point feature triplets,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 648–663 (cit. on p. 13).

[140] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571 (cit. on pp. 13, 14).

[141] H. Zhan, C. S. Weerasekera, J.-W. Bian, and I. Reid, “Visual odometry revisited: What should be learnt?” In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 4203–4210 (cit. on p. 15).

[142] P. V. Gakne and K. O’Keefe, “Tackling the scale factor issue in a monocular visual odometry using a 3d city model,” *Proceedings of the ITSNT*, 2018 (cit. on p. 15).

[143] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105 (cit. on p. 15).

[144] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2 (cit. on p. 16).

[145] M. W. Berry, A. Mohamed, and B. W. Yap, *Supervised and unsupervised learning for data science*. Springer, 2019 (cit. on p. 16).

[146] J. Tang, L. Ericson, J. Folkesson, and P. Jensfelt, “Gcnv2: Efficient correspondence prediction for real-time slam,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3505–3512, 2019 (cit. on p. 16).

[147] M. Dusmanu, O. Miksik, J. L. Schönberger, and M. Pollefeys, “Cross-descriptor visual localization and mapping,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6058–6067 (cit. on p. 16).

[148] L. Cavalli, V. Larsson, M. R. Oswald, T. Sattler, and M. Pollefeys, “Adalam: Revisiting handcrafted outlier detection,” *arXiv preprint arXiv:2006.04250*, 2020 (cit. on p. 16).

[149] P. Agrawal, J. Carreira, and J. Malik, “Learning to see by moving,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 37–45 (cit. on p. 17).

[150] S. J. Lee, H. Choi, and S. S. Hwang, “Real-time depth estimation using recurrent cnn with sparse depth cues for slam system,” *International Journal of Control, Automation and Systems*, vol. 18, no. 1, pp. 206–216, 2020 (cit. on p. 17).

[151] P. Marquez Valle, *A confidence framework for the assessment of optical flow performance*. Universitat Autonoma de Barcelona, 2015 (cit. on p. 17).

[152] C.-H. Lee and Z. Wu, “Optical flow estimation by integrating feature-based with flow-based schemes under multiresolution,” 1989 (cit. on p. 17).

[153] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997 (cit. on p. 18).

[154] F. Walch, C. Hazirbas, L. Leal-Taixe, *et al.*, "Image-based localization using lstms for structured feature correlation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 627–637 (cit. on p. 18).

[155] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 822–838 (cit. on p. 21).

[156] T.-W. Hui, X. Tang, and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8981–8989 (cit. on pp. 22, 24).

[157] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," 2019 (cit. on pp. 22, 24, 55, 62).

[158] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005 (cit. on p. 22).

[159] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i," *IEEE robotics & automation magazine*, vol. 13, no. 2, pp. 99–110, 2006 (cit. on p. 24).

[160] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 3354–3361 (cit. on pp. 24, 29, 30, 53, 54, 56).

[161] Y. Almalioglu, A. Santamaria-Navarro, B. Morrell, and A.-a. Agha-mohammadi, "Unsupervised deep persistent monocular visual odometry and depth estimation in extreme environments," *arXiv preprint arXiv:2011.00341*, 2020 (cit. on p. 25).

[162] M. Paulraj and C. Hema, "Vision based systems for localization in service robots," *Robot Localization and Map Building*, p. 309, 2010 (cit. on p. 25).

[163] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. arXiv: 1701.08376 (cit. on p. 26).

[164] D. Fernandez and A. Price, "Visual odometry for an outdoor mobile robot," in *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, IEEE, vol. 2, 2004, pp. 816–821 (cit. on p. 26).

[165] I. Durazo-Cardenas, A. Starr, A. Tsourdos, M. Bevilacqua, and J. Morineau, "Precise vehicle location as a fundamental parameter for intelligent selfaware rail-track maintenance systems," *Procedia CIRP*, vol. 22, no. 1, pp. 219–224, 2014 (cit. on p. 26).

[166] D. Wang, H. Liang, H. Zhu, and S. Zhang, "A bionic camera-based polarization navigation sensor," *Sensors*, vol. 14, no. 7, pp. 13 006–13 023, 2014 (cit. on p. 26).

[167] B. Barshan and H. F. Durrant-Whyte, "An inertial navigation system for a mobile robot," *IFAC Proceedings Volumes*, vol. 26, no. 1, pp. 54–59, 1993 (cit. on p. 26).

[168] S. Royo and M. Ballesta-Garcia, "An overview of lidar imaging systems for autonomous vehicles," *Applied Sciences*, vol. 9, no. 19, p. 4093, 2019 (cit. on p. 26).

[169] R. Thakur, “Scanning lidar in advanced driver assistance systems and beyond: Building a road map for next-generation lidar technology,” *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 48–54, 2016 (cit. on p. 26).

[170] M. Kutila, P. Pyykönen, W. Ritter, O. Sawade, and B. Schäufele, “Automotive lidar sensor development scenarios for harsh weather conditions,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016, pp. 265–270 (cit. on p. 26).

[171] J. Horn and G. Schmidt, “Continuous localization of a mobile robot based on 3d-laser-range-data, predicted sensor images, and dead-reckoning,” *Robotics and Autonomous Systems*, vol. 14, no. 2-3, pp. 99–118, 1995 (cit. on p. 26).

[172] K. Lingemann, A. Nüchter, J. Hertzberg, and H. Surmann, “High-speed laser localization for mobile robots,” *Robotics and autonomous systems*, vol. 51, no. 4, pp. 275–296, 2005 (cit. on p. 26).

[173] J. Borenstein, H. Everett, L. Feng, *et al.*, “Where am i? sensors and methods for mobile robot positioning,” *University of Michigan*, vol. 119, no. 120, p. 27, 1996 (cit. on p. 27).

[174] F. Potorti, S. Park, A. R. Jimenez Ruiz, *et al.*, “Comparing the performance of indoor localization systems through the evaal framework,” *Sensors*, vol. 17, no. 10, p. 2327, 2017 (cit. on p. 27).

[175] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, “Real-time rgb-d camera relocalization,” in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2013, pp. 173–179 (cit. on p. 30).

[176] Y. Li, N. Snavely, and D. P. Huttenlocher, “Location recognition using prioritized feature matching,” in *European conference on computer vision*, Springer, 2010, pp. 791–804 (cit. on p. 30).

[177] A. Handa, T. Whelan, J. McDonald, and A. Davison, “A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM,” in *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, 2014 (cit. on p. 30).

[178] S. Cortés, A. Solin, E. Rahtu, and J. Kannala, “Advio: An authentic dataset for visual-inertial odometry,” in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., Cham: Springer International Publishing, 2018, pp. 425–440 (cit. on p. 30).

[179] D. Zuñiga-Noël, A. Jaenal, R. Gomez-Ojeda, and J. Gonzalez-Jimenez, “The uma-vi dataset: Visual–inertial odometry in low-textured and dynamic illumination environments,” *The International Journal of Robotics Research*, vol. 39, no. 9, pp. 1052–1060, 2020. eprint: `https://doi.org/10.1177/0278364920938439` (cit. on p. 30).

[180] S. Ceriani, G. Fontana, A. Giusti, *et al.*, “Rawseeds ground truth collection systems for indoor self-localization and mapping.,” *Auton. Robots*, vol. 27, no. 4, pp. 353–371, Dec. 17, 2009 (cit. on p. 30).

[181] J. Xiao, A. Owens, and A. Torralba, “Sun3d: A database of big spaces reconstructed using sfm and object labels,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1625–1632 (cit. on p. 30).

[182] S. Klenk, J. Chui, N. Demmel, and D. Cremers, “Tum-vie: The tum stereo visual-inertial event dataset,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2021. arXiv: `2108.07329 [cs.CV]` (cit. on p. 30).

[183] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, 2012 (cit. on p. 30).

[184] J. Engel, V. Usenko, and D. Cremers, “A photometrically calibrated benchmark for monocular visual odometry,” in *arXiv:1607.02555*, 2016 (cit. on p. 30).

[185] M. Fallon, H. Johannsson, M. Kaess, and J. J. Leonard, “The mit stata center dataset,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1695–1699, 2013 (cit. on p. 30).

[186] H. Alismail, B. Browning, and M. B. Dias, “Evaluating pose estimation methods for stereo visual odometry on robots,” in *the 11th Int’l Conf. on Intelligent Autonomous Systems (IAS-11)*, vol. 3, 2010, p. 2 (cit. on pp. 30, 53).

[187] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 573–580 (cit. on pp. 30, 53).

[188] T. Schops, J. L. Schonberger, S. Galliani, *et al.*, “A multi-view stereo benchmark with high-resolution images and multi-camera videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3260–3269 (cit. on p. 30).

[189] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, “University of Michigan North Campus long-term vision and lidar dataset,” *International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2015 (cit. on p. 30).

[190] D. Olid, J. M. Fácil, and J. Civera, “Single-view place recognition under seasonal changes,” in *PPNIV Workshop at IROS 2018*, 2018 (cit. on pp. 30, 31, 90).

[191] A. L. Majdik, C. Till, and D. Scaramuzza, “The zurich urban micro aerial vehicle dataset,” *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 269–273, 2017. eprint: `https://doi.org/10.1177/0278364917702237` (cit. on p. 30).

[192] A. Z. Zhu, D. Thakur, T. Ozaslan, *et al.*, “The multivehicle stereo event camera dataset: An event camera dataset for 3d perception,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018 (cit. on p. 30).

[193] M. T. Ohradzansky, E. R. Rush, D. G. Riley, *et al.*, “Multi-agent autonomy: Advancements and challenges in subterranean exploration,” *arXiv preprint arXiv:2110.04390*, 2021 (cit. on p. 31).

[194] ÖPNVKarte map, *Planet dump retrieved from https://planet.osm.org*, 2017 (cit. on pp. 34, 43).

[195] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017. arXiv: `arXiv:1610.06475v2` (cit. on p. 39).

[196] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 796–803, 2017 (cit. on p. 40).

[197] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, pp. 1–11, 2017 (cit. on p. 40).

[198] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 2502–2509 (cit. on p. 40).

[199] Google Maps, *Directions for Driving in Street Aragoa and Musakola, Arrasate/Mondragon*, 2021 (cit. on p. 40).

[200] H. Zhao, B. Zhang, C. Wu, Z. Zuo, and Z. Chen, "Development of a coordinate transformation method for direct georeferencing in map projection frames," *ISPRS journal of photogrammetry and remote sensing*, vol. 77, pp. 94–103, 2013 (cit. on p. 43).

[201] OpenStreetMap contributors, *Planet dump retrieved from https://planet.osm.org*, 2017 (cit. on p. 43).

[202] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004 (cit. on pp. 48, 49).

[203] Z. Ni, W. Yang, S. Wang, L. Ma, and S. Kwong, "Towards unsupervised deep image enhancement with generative adversarial network," *IEEE Transactions on Image Processing*, vol. 29, pp. 9140–9151, 2020 (cit. on p. 48).

[204] Y. Gao, A. Rehman, and Z. Wang, "Cw-ssim based image classification," in *2011 18th IEEE International Conference on Image Processing*, IEEE, 2011, pp. 1249–1252 (cit. on p. 48).

[205] J. Søgaard, L. Krasula, M. Shahid, *et al.*, "Applicability of existing objective metrics of perceptual quality for adaptive video streaming," *Electronic Imaging*, vol. 2016, no. 13, pp. 1–7, 2016 (cit. on p. 48).

[206] Y. Jiang, Y. Xu, and Y. Liu, "Performance evaluation of feature detection and matching in stereo visual odometry," *Neurocomputing*, vol. 120, pp. 380–390, 2013 (cit. on p. 53).

[207] M. Grupp, *Evo: Python package for the evaluation of odometry and slam.* `https://github.com/MichaelGrupp/evo`, 2017 (cit. on p. 53).

[208] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991 (cit. on p. 54).

[209] H. Zhan, C. S. Weerasekera, J. Bian, and I. Reid, *Df-vo*, `https://github.com/Huangying-Zhan/DF-VO`, 2021 (cit. on p. 55).

[210] A. Dosovitskiy, P. Fischer, E. Ilg, *et al.*, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766 (cit. on p. 55).

[211] H. Zhan, C. S. Weerasekera, J. Bian, and I. Reid, "Visual odometry revisited: What should be learnt?" *arXiv preprint arXiv:1909.09803*, 2019 (cit. on pp. 62, 91).

[212] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8981–8989 (cit. on p. 62).

[213] R. Maini and H. Aggarwal, "A comprehensive review of image enhancement techniques," *arXiv preprint arXiv:1003.4053*, 2010 (cit. on p. 64).

[214] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019 (cit. on pp. 64–66).

[215] A. Jurio, M. Pagola, M. Galar, C. Lopez-Molina, and D. Paternain, "A comparison study of different color spaces in clustering based image segmentation," in *International conference on information processing and management of uncertainty in knowledge-based systems*, Springer, 2010, pp. 532–541 (cit. on p. 64).

[216] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014 (cit. on p. 64).

[217] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, "Deep image: Scaling up image recognition," *arXiv preprint arXiv:1501.02876*, vol. 7, no. 8, 2015 (cit. on p. 64).

[218] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*, IEEE, 2018, pp. 117–122 (cit. on p. 64).

[219] G. Kang, X. Dong, L. Zheng, and Y. Yang, "Patchshuffle regularization," *arXiv preprint arXiv:1707.07103*, 2017 (cit. on p. 64).

[220] F. J. Moreno-Barea, F. Strazzera, J. M. Jerez, D. Urda, and L. Franco, "Forward noise adjustment scheme for data augmentation," in *2018 IEEE symposium series on computational intelligence (SSCI)*, IEEE, 2018, pp. 728–734 (cit. on p. 64).

[221] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012 (cit. on p. 64).

[222] R. Takahashi, T. Matsubara, and K. Uehara, "Data augmentation using random image cropping and patching for deep cnns," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, pp. 2917–2931, 2019 (cit. on p. 64).

[223] H. Inoue, "Data augmentation by pairing samples for images classification," *arXiv preprint arXiv:1801.02929*, 2018 (cit. on p. 64).

[224] C. Summers and M. J. Dinneen, “Improved mixed-example data augmentation,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 1262–1270 (cit. on p. 64).

[225] C. Szegedy, W. Zaremba, I. Sutskever, *et al.*, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013 (cit. on p. 65).

[226] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2014 (cit. on p. 65).

[227] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014 (cit. on p. 65).

[228] D. Mahapatra and B. Bozorgtabar, “Retinal vasculature segmentation using local saliency maps and generative adversarial networks for image super resolution,” *arXiv preprint arXiv:1710.04783*, 2017 (cit. on p. 65).

[229] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232 (cit. on pp. 65, 66).

[230] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015 (cit. on p. 65).

[231] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014 (cit. on p. 65).

[232] M.-Y. Liu and O. Tuzel, “Coupled generative adversarial networks,” *Advances in neural information processing systems*, vol. 29, 2016 (cit. on p. 65).

[233] J. Donahue, P. Krähenbühl, and T. Darrell, “Adversarial feature learning,” *arXiv preprint arXiv:1605.09782*, 2016 (cit. on p. 66).

[234] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134 (cit. on p. 66).

[235] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation generative adversarial networks,” *arXiv preprint arXiv:1711.04340*, 2017 (cit. on p. 66).

[236] C. Guo, C. Li, J. Guo, *et al.*, “Zero-reference deep curve estimation for low-light image enhancement,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1780–1789 (cit. on p. 66).

[237] S. W. Zamir, A. Arora, S. Khan, *et al.*, “Learning enriched features for real image restoration and enhancement,” in *European Conference on Computer Vision*, Springer, 2020, pp. 492–511 (cit. on p. 66).

[238] F. Lv, Y. Li, and F. Lu, “Attention guided low-light image enhancement with a large scale low-light simulation dataset,” *arXiv preprint arXiv:1908.00682*, 2019 (cit. on p. 66).

[239] C. Wei, W. Wang, W. Yang, and J. Liu, “Deep retinex decomposition for low-light enhancement,” *arXiv preprint arXiv:1808.04560*, 2018 (cit. on pp. 66, 67).

[240] E. Jung, N. Yang, and D. Cremers, “Multi-frame gan: Image enhancement for stereo visual odometry in low light,” in *Conference on Robot Learning*, PMLR, 2020, pp. 651–660 (cit. on p. 66).

[241] X. Guo, “Lime: A method for low-light image enhancement,” in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 87–91 (cit. on p. 66).

[242] Y. Jiang, X. Gong, D. Liu, *et al.*, “Enlightengan: Deep light enhancement without paired supervision,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2340–2349, 2021 (cit. on pp. 66, 67, 69, 72, 92).

[243] J. Deng, W. Dong, R. Socher, *et al.*, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255 (cit. on p. 67).

[244] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, “Raise: A raw images dataset for digital image forensics,” in *Proceedings of the 6th ACM multimedia systems conference*, 2015, pp. 219–224 (cit. on p. 67).

[245] N. K. Kalantari, R. Ramamoorthi, *et al.*, “Deep high dynamic range imaging of dynamic scenes.,” *ACM Trans. Graph.*, vol. 36, no. 4, pp. 144–1, 2017 (cit. on p. 67).

[246] J. Cai, S. Gu, and L. Zhang, “Learning a deep single image contrast enhancer from multi-exposure images,” *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 2049–2062, 2018 (cit. on p. 67).

[247] L. Zhu, H. Luo, and X. Zhang, “Uncertainty and sensitivity analysis for camera calibration,” *Industrial Robot: An International Journal*, 2009 (cit. on p. 67).

[248] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000 (cit. on p. 68).

[249] D. G. Altman and J. M. Bland, “Standard deviations and standard errors,” *Bmj*, vol. 331, no. 7521, p. 903, 2005 (cit. on p. 70).

[250] C. S. Rayat, “Measures of dispersion,” in *Statistical Methods in Medical Research*, Springer, 2018, pp. 47–60 (cit. on p. 86).

Publications A

A.1 Application of Computer Vision and Deep Learning in the railway domain for autonomous train stop operations

This paper was presented in the International Symposium on System Integration (SII) in 2020, and then published in the conference proceedings. The full citation:

Etxeberria-Garcia, M., Labayen, M., Zamalloa, M., and Arana-Arexolaleiba, N. (2020, January). Application of Computer Vision and Deep Learning in the railway domain for autonomous train stop operation. In *2020 IEEE/SICE International Symposium on System Integration (SII) (pp. 943-948). IEEE.*

Proceedings of the 2020 IEEE/SICE
International Symposium on System Integration
Honolulu, Hawaii, USA, January 12-15, 2020

Application of Computer Vision and Deep Learning in the railway domain for autonomous train stop operation

Mikel Etxeberria-Garcia[1], Mikel Labayen[2], Maider Zamalloa[1] and Nestor Arana-Arexolaleiba[3]

Abstract— **The purpose of this paper is to present the results of the analysis of the application of Deep Learning in the railway domain with a particular focus on a train stop operation. The paper proposes an approach consisting of monocular vision-based and Deep Learning architectures. Even the difficulties imposed by actual regulation, the findings show that Deep Learning architecture can offer promising results in railway localization using techniques like visual odometry, SLAM or pose estimation. Besides, in spite of the many datasets available in the literature needed to train the neural network, none of them have been created for indoor railway environments. Therefore, a new dataset should be created. Furthermore, the paper presents future research and development suggestions for railway applications which contribute to guiding the mid-term research and development.**

I. INTRODUCTION

The application of Machine Learning has increased since the applicability of some of its techniques has improved. Deep Learning is one of the most growing techniques of Machine Learning. The good results that have achieved in recent researches and the increase of computational capacity have lead to a time where Deep Learning can be applied to a wide range of domains [1]. From Medical technologies to the Internet of Things through its mayor domain, robotics. The power of Deep Learning in robotics lies in the potential it has of making a system that can learn [2]. The robotics community has identified and summarized several applications for Deep Learning in robotics, such as, learning complex dynamics, control operations, advanced manipulation, object recognition or interpretation of human actions [3].

In this context, Deep Learning application has facilitated a development in the autonomous driving industry as one of the most important future business bets. Computer vision techniques, using Deep Learning, have helped to create machine-learning-based robots and cars that can predict and learn how to drive in various environments. The recent advances on Intelligent Transportation Systems (ITS), Advanced Driving Assistance Systems (ADAS), intelligent infrastructures and autonomous driving have carried many benefits to the transportation industry [4]. These technologies provide the vehicle its own decision-making capacity and the ability to interpret its environment, and consequently, enhance the control and signaling solutions. The irruption of Artificial Intelligence techniques in general and Deep Learning techniques, in particular, have allowed improving the perception capacity of these systems and the knowledge derived from the information perceived in the environment.

The railway domain is also transforming towards the ITS and ADAS industry. Nowadays, this sector is ready for the next steps involving itself in different research projects related to Computer Vision and Artificial Intelligence development. In a fully autonomous train system all the operations involved in must be automatic, for example, visual odometry, people and obstacle detection-identification in railroads, operations such as train doors opening/closing, gauge control in platforms, coupling or as is presented in this work, accurate train stopping in train platforms.

This paper is divided into the following sections. Section II presents a use case of a company related to the railway domain. Main approaches in train-robot localization using Deep Learning are explored in Section III, followed by the first use case presentation in Section IV. Finally, some expected results are drawn in Section V.

II. PROBLEM DEFINITION

Communication-Based Train Control (CBTC) is a standard defined by the IEEE (IEEE 1474 [5]) which defines a set of performance and functional requirements for track and onboard equipment in order to enhance performance, availability, operations and the protection of the involved systems. A CBTC system could be defined as an automatic train control system where the track and onboard subsystems are continuously communicated. The main two functionalities covered by those subsystems are the Automatic Train Protection (ATP) and Automatic Train Operation (ATO). ATP subsystems monitor the train speed and position in order to guarantee a safe train operation. On the other hand, ATO subsystems are dedicated to the operations devoted to reaching a more autonomous and efficient train driving experience, such as, driving assistance tasks or automatic control of train brake and traction commands that aim to ensure that train speed is lower than the limit established by the ATP system [6].

Current CBTC systems, according to the standard IEC 62290-1, can be divided into pre-established Grades of Autonomy (GOA). The GOA of a train implementing any autonomous operation will have a value between 2 and 4: GOA2 for a semi-automated Train Operation, GOA3 for a driverless Train Operation and GOA4 for an unattended Train

[1]Ikerlan Technology Research Centre, Dependable Embedded Systems Area, P. J. M. Arizmendiarrieta, 2 20500 Arrasate-Mondragon, Gipuzkoa, Spain {mikel.etxeberria, mzamalloa}@ikerlan.es

[2]Faculty of Informatics, UPV/EHU, Manuel Lardizabal Ibilbidea, 1, 20018 Donostia, Gipuzkoa, Spain mikel.labayen@ehu.eus

[3]MGEP, Mondragon Unibertsitatea, Loramendi Kalea, 4 20500 Arrasate-Mondragon Gipuzkoa, Spain narana@mondragon.edu

This research is supported by Basque Government through BIKAINTEK2018 program and CAF Signalling. In collaboration between Ikerlan, CAF Signalling and Mondragon Unibertsitatea.

978-1-7281-6667-4/20/$31.00 ©2020 IEEE

Operation. In GOA3 and GOA4 systems, as there is not a driver inside the train, an accurate train location system is required. Precise positioning systems can reach a higher grade of automation [7]. A train that implements GOA3 or GOA4 level can be considered as a robot that navigates through a track in indoor and outdoor environments including underground stations. Therefore, it becomes essential to implement precise and reliable train localization subsystems. A GOA3 or GOA4 train must compute, among others, the braking curve or the train stopping location with precision.

Accurate train localization and platform-train doors alignment are required for a safe passenger transfer train operation. Door equipped platforms, which avoid human or undesirable objects to fall in the railway area, are more common now than in the past. To align the doors and platform in a train stopping point requires a precise localization information. Nowadays, it is calculated using train speed data captured form different odometry sensors. These sensor errors are corrected time-to-time during train service using beacon information. However, in a stopping point, the driver's eyes and experience are still the key factors to align correctly the train with the platform area and to remove final localization error.

The sensors used in visual localization systems include monocular, stereo, RGB-D cameras, and LIDAR. In visual problems, some cameras are not able to calculate the absolute scale, and therefore, scale drifts appear [8]. Stereo cameras provide an immediate scale while requiring calibration. RGB-D cameras provide color and depth information for each pixel in an image [9], but its economical cost is higher than the other options. In general, a lot of research interest has been focused on dense and semi-dense methods from a single camera [10]. In the railway domain, the only research found using a monocular camera is DisNet [11] proposed by Haseeb *et al.* Most approaches in this domain are based on another type of sensors as stereo cameras [12], [13].

The main objective of this research is to explore the capability of Machine Learning techniques, particularly Deep Learning techniques, and monocular computer vision for an accurate train stopping in fully autonomous train stop operation.

III. RELATED WORK

Lately, the capacity of Computer Vision to address some robotics problems has increased due to the rise of the application of Deep Learning algorithms and the increase of computational resources. This situation also comes from the promising results obtained by the application of deep approaches in computer vision, specifically with the use of Deep Neural Networks (DNN), as Convolutional Neural Networks (CNN), on large-scale image classification (Krizhevsky *et al.* [14]). This work demonstrates the idea of the benefits of using CNNs on Computer Vision problems. Additionally, it has been shown that one of the potentials of CNN is their generalization ability in visual recognition tasks, i.e., visual localization estimation. A CNN trained for another purpose at first instance can be reused to solve another purpose without the need for a full training phase again. Most systems use CNNs to find only local features or generate descriptors of discrete proposal regions [15]. Several works state that deep learning algorithms can model localization or depth solutions by regression [16]. On the railway domain, most researches focus on other computer vision problems as object and rail detection [17], [18] or stations monitoring [19] although some of these approaches may be applied for localization purposes.

Three main techniques can be distinguished in visual localization problems: Visual Odometry (VO), Simultaneous Location and Mapping (SLAM) and Depth Estimation. Some of these techniques refer to the same problems, share viewpoints and in some cases cannot be differentiated.

- *Visual Odometry (VO).* Odometry can be defined as the use of data from motion sensors in order to estimate changes in position over time [20]. Visual odometry (VO) is a particular case of odometry, where the position information is acquired through camera images [8]. The term Visual Odometry was first introduced by Niester *et al.* [21] proposing a method for estimating camera motion using RANSAC [22] outlier refinement method and tracking extracted features across all frames. Before that, feature matching was done just in consecutive frames. Later researches have shown that VO methods perform significantly better than wheel odometry in robotics while the cost of cameras is much lower compared to more accurate IMUs and LASER scanners [8]. This scenario raises the need for exploration of the applicability of VO in the railway domain and autonomous driving trains.
- *SLAM.* Simultaneous Localization and Mapping (SLAM) is a technique to reconstruct an unknown 3D environment. It has become a popular research topic, as it is the base for autonomous robot navigation. Visual SLAM (vSLAM) is the field of SLAM comprised of methods that use visual information. Both share many components such as feature extractors. The main difference between both techniques is that VO centers on a relative part of the map, while vSLAM uses the full context and global consistency is aimed [23].
- *Depth estimation.* Scene depth refers to the distance from the camera optical center to the object along to the optical axis [24]. The estimation of depth can contribute to localization, and in many approaches is used as a SLAM phase.

Some recent works based on previously mentioned techniques, apply deep learning algorithms in VO solutions. They can estimate the pose directly from an input image without feature extraction or feature matching processes. In [25], Kendall *et al.* proposed PoseNet, a robust and real-time monocular re-localization system based on an end-to-end trained CNN. This approach was improved by adding a treatment of scene geometry introducing geometric loss functions [26]. Wang *et al.* [27] presented DeepVO, an approach that mixes CNN and RNN called Recurrent Convolutional Neural

Network (RCNN). It takes the benefits of both networks, the feature extraction capabilities of the CNN and the sequential modeling from the RNN. Besides, Clark *et al.* [16] extended PoseNet with an RNN in order to exploit temporal dependencies and improve the monocular localization accuracy. Some approaches that extend the PoseNet system have been presented, i.e., relative ego-motion [28]. Later, in [29] Xiang *et al.* introduced PoseCNN, a CNN for 6D object pose estimation. PoseCNN localizes an object center in the image and predicts its distance from the camera.

From all the explored techniques, three of them have been selected as the most interesting and relevant for our particular use case:

- Disnet. It is the only approach based on the same domain and is based on CNNs. It uses a CNN to regress the distance to previously detected objects by a monocular camera installed on a train. For the object detection part uses the standard YOLO [30] algorithm, also based on CNNs.
- PoseCNN. Detects the center of a known object and estimates the distance from the camera to regress the pose of that object using a CNN.
- DeepVO. Introduces Recurrent Networks to the localization problem and takes advantage of the input videos as it infers poses of objects directly from a sequence of images.

The application of these techniques is foreseen in a real-world use case from the railway environment, as there are few applications of deep learning approaches in this domain due to strict railway regulation. The main goal is to explore the applicability of Deep Learning for Visual Odometry, SLAM, and Depth estimation in the railway domain.

IV. USE CASE: TRAIN STOP OPERATION

A. Use case definition

The use case scenario is the Autonomous Urban Train where artificial intelligence and high-performance computational capabilities are used to increase the dependability and the safety of the system. The objective is to apply Computer Vision and Deep Learning techniques to improve different autonomous train operation functionalities as precision stop, rolling stock coupling operation or person and obstacle detection-identification in railroads.

The selected use case is the automatic accurate stop at door equipped platforms aligning the vehicle and platform doors. The goal is to perform precise localization inside the platform area using visual patterns detection, identification and tracking in order to reach an accurate stopping point and managing automatic train operation (traction and brake commands, ATO functionality). A contribution is expected to the automatic train operation system, adding the visual localization estimation information to the usual trains odometry data calculations based on radars and encoders.

In the current train localization system, beacon positions are known by trackside equipment and may be known by train if previously announced. From beacon to beacon, a localization error is accumulated that is proportional to traveled distance. Each time the train crosses a beacon, the localization and accuracy are reset. With the combination of wheel odometry data, given by radars and encoders, and Visual Odometry (VO) data, provided by our proposed approaches, an improvement on the precision of the stop is foreseen, where the localization error must be lower than the current error given by beacon-based train localization system.

The main idea of our approach is to detect a pattern that is always placed on the platform that will help us to locate the train through Deep Neuronal Networks (DNN). These patterns usually are used by train drivers to know the stopping position of the train and have a regular form and color. One example is shown in figure 1.

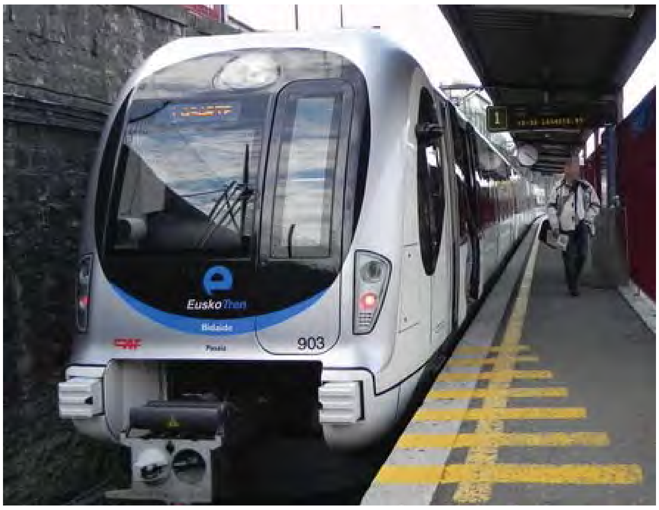


Fig. 1. Signaling patterns are placed at the end of the platforms, as the yellow pattern shown in this figure.

B. Architecture

The architecture of the designed application for this use case is shown in figure 2. The videos are captured using a camera that transfers the images to a capturer, which has two workflows. First, transfers the videos to a database (DB) that will be used to pre-process the videos and train a DNN. The training process of the DNN will produce a model that will be used later for real time processing. Secondly, the capturer passes the streaming of frames to the previously trained DNN that will output the desired result. Depending on the selected approach, the pre-process done to the input videos, the structure of the DNN and the output will be different. Usually, the pre-processing phase is done using Computer Vision techniques without Machine Learning.

C. Datasets and data collection

Deep Learning approaches require large amounts of data for training. This data can be collected from different sources: use data from existing real datasets and using simulated environments. The first option is to use data from the standard datasets created by other institutions or researches previously or, in case those datasets do not fit the

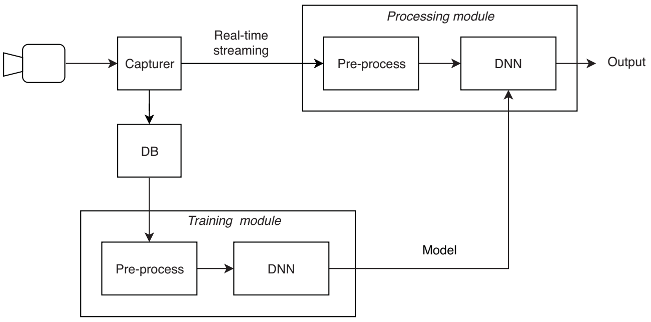


Fig. 2. The architecture of the designed application for train localization using Computer Vision and Deep Neural Networks (DNN)

problem, to create a new dataset. Having a properly labeled ground truth in these datasets is essential as they are the base for training DNNs and evaluating performances. Depending on the selected approach the required ground truth data is not the same. After an analysis of the most used databases for visual localization researches, we have found that only one database (Norland [37]) covers the railway domain. Furthermore, it does not match our needs for different type of scenarios, including indoor environments. A summary of database analysis is shown in table I. For each database, the used sensors, the domain it belongs to, if they give pose or/and depth information and if it is an indoor or an outdoor research is addressed.

In the case of new datasets, an appropriate environment is required, where a camera can be used to take images from the front of the train to the track. The advantage of this system is that the database can be designed to the particular use case, but it requires a lot of time of recording for experimentation in the track and means to generate a ground truth. To overcome this problem, simulated environments can be used, where no real railways are involved. The drawback of simulated environments is that we can not assure that an algorithm trained and validated in a simulated environment will give the same results in a real world scenario.

Therefore, the creation of a database is envisioned to afford the lack of a standard dataset that fits this research. The database will be collected in indoor stations including scenarios with unfavorable light conditions, pattern shape/color degradation due to the passing of time and partial occlusions (hidden patterns because of people or object presence).

V. Expected Results and Conclusions

The main goal of this research is to explore the applicability of the Visual Odometry and Deep Learning techniques used in robotics and autonomous car vehicles to the railway domain. This article presents a train stopping use case based on improving the train localization estimation in indoor environments. It also defines the main architecture of the system designed to solve the presented use case. Finally, the need for a dataset oriented to validate and test indoor train visual localization systems is pointed.

As the results, a visual localization system that improves the accuracy of indoor localization systems is expected. According to the accuracy requirements, localization error should be lower than 15 cm at 99,9% of times (measure errors must to be taken into account). Consequently, the system should be able to perform an accurate automatic stop at door equipped platforms, aligning the vehicle and platform for correct passenger transfer.

Railway operators are interested in more accessible market and flexible solutions aligned with social sustainability and mobility concerns. If urban vehicles (metro) gain autonomy, system development cost is reduced (install and maintenance costs) and operation flexibility is gained. The information received from railroad signaling modes can be enriched by computer vision and deep learning approaches giving to vehicles more autonomy and decision-making capabilities. This way they can observe and interpret the environment in an independent manner.

Dataset	Domain	Sensors	Pose	Depth	Indoor/Outdoor
SUN3D [31]	Robot	RGB-D camera	X	X	I
TUM-LSI [32]	Robot	RGB-D camera	X		I
NavVis [32]	Robot	Camera	X		I
Cambridge [25]	Robot	Smartphone	X		O
7-scenes [33]	Urban localization	RGB-D camera	X		I
BigSFM [34]	Urban localization	Camera	X		O
MIT DATA [35]	Robot	LIDAR, Stereo camera, Odometry	X		I
KITTI [36]	Car	Laser, Stereo camera, GPS	X	X	O
Nordland [37]	**Train**	**Camera, GPS/INS**	**X**		**O**
Oxford RobotCar [38]	Car	Cameras, LIDAR, GPS/INS	X		O
EuroC/MAV [39]	Micro Aerial Vehicle	Stereo camera, Laser, IMU	X	X	I
The Wean Hall [40]	Robot	Stereo camera, Laser, IMU	X	X	I
Ford Campus [41]	Car	Camera, LIDAR, IMU	X		O
RGB-D SLAM [42]	Robot	RGB-D camera, Accelerometer	X	X	I
GMU Kitchen [43]	3D reconstruction	RGB-D camera	X	X	I
NYUD/NYUD2 [44]	3D reconstruction	RGB-D camera		X	I
ETH3D [45]	3D reconstruction	Camera, Laser, IMU	X	X	I/O
Make3D [46]	3D reconstruction	Camera, Laser	X	X	I/O
MPI-Sintel [47]	Movie	(Digital)		X	I/O

TABLE I

Most used datasets for visual localization problems

REFERENCES

[1] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[2] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, “Building machines that learn and think like people,” *Behavioral and brain sciences*, vol. 40, 2017.

[3] H. A. Pierson and M. S. Gashler, “Deep learning in robotics: a review of recent research,” *Advanced Robotics*, vol. 31, no. 16, pp. 821–835, 2017.

[4] W. Bernhart and M. Winterhoff, “Autonomous driving: Disruptive innovation that promises to change the automotive industry as we know it,” in *Energy Consumption and Autonomous Driving*. Springer, 2016, pp. 3–10.

[5] “Ieee standard for communications-based train control (cbtc) performance and functional requirements,” *IEEE Std 1474.1-2004 (Revision of IEEE Std 1474.1-1999)*, pp. 1–45, 2004.

[6] M. Malvezzi, B. Allotta, and M. Rinchi, “Odometric estimation for automatic train protection and control systems,” *Vehicle System Dynamics*, vol. 49, no. 5, pp. 723–739, 2011.

[7] T. Albrecht, K. Lüddecke, and J. Zimmermann, “A precise and reliable train positioning system and its use for automation of train operation,” *IEEE ICIRT 2013 - Proceedings: IEEE International Conference on Intelligent Rail Transportation*, pp. 134–139, 2013.

[8] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, “An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics,” *Intelligent Industrial Systems*, vol. 1, no. 4, pp. 289–311, 2015.

[9] S. Poddar, R. Kottath, and V. Karar, “Evolution of Visual Odometry Techniques,” 2018. [Online]. Available: http://arxiv.org/abs/1804.11142

[10] K. Tateno, F. Tombari, I. Laina, and N. Navab, “CNN-SLAM : Real-time dense monocular SLAM with learned depth prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6243–6252.

[11] M. A. Haseeb, J. Guan, D. Ristić-Durrant, and A. Gräser, “Disnet: A novel method for distance estimation from monocular camera,” 2012.

[12] J. Weichselbaum, C. Zinner, O. Gebauer, and W. Pree, “Accurate 3d-vision-based obstacle detection for an autonomous train,” *Computers in Industry*, vol. 64, no. 9, pp. 1209–1220, 2013.

[13] N. Fakhfakh, L. Khoudour, E.-M. El-Koursi, J.-L. Bruyelle, A. Dufaux, and J. Jacot, “Background subtraction and 3d localization of moving and stationary obstacles at level crossings,” in *2010 2nd International Conference on Image Processing Theory, Tools and Applications*. IEEE, 2010, pp. 72–78.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] D. Eigen and R. Fergus, “Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[16] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen, “VidLoc : A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6856–6864.

[17] M. Karakose, O. Yaman, M. Baygin, K. Murat, and E. Akin, “A new computer vision based method for rail track detection and fault diagnosis in railways,” *International Journal of Mechanical Engineering and Robotics Research*, vol. 6, no. 1, pp. 22–17, 2017.

[18] S. Mittal and D. Rao, “Vision based railway track monitoring using deep learning,” *arXiv preprint arXiv:1711.06423*, 2017.

[19] S. Oh, S. Park, and C. Lee, “Vision based platform monitoring system for railway station safety,” in *2007 7th International Conference on ITS Telecommunications*. IEEE, 2007, pp. 1–5.

[20] J. Otegui, A. Bahillo, I. Lopetegi, and L. E. Diez, “A Survey of Train Positioning Solutions,” *IEEE Sensors Journal*, vol. 17, no. 20, pp. 6788–6797, 2017.

[21] D. Nistér, O. Naroditsky, and J. Bergen, “Visual Odometry,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004, p. 1.

[22] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] T. Taketomi, H. Uchiyama, and S. Ikeda, “Visual slam algorithms: A survey from 2010 to 2016,” *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017.

[24] L. He, G. Wang, and Z. Hu, “Learning depth from single images with deep neural network embedding focal length,” *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4676–4689, 2018.

[25] A. Kendall, M. Grimes, and R. Cipolla, “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[26] A. Kendall and R. Cipolla, “Geometric loss functions for camera pose regression with deep learning,” *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6555–6564, 2017.

[27] S. Wang, R. Clark, H. Wen, and N. Trigoni, “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2043–2050.

[28] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, “Relative camera pose estimation using convolutional neural networks,” in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2017, pp. 675–687.

[29] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, “Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes,” *arXiv preprint arXiv:1711.00199*, 2017.

[30] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[31] J. Xiao, A. Owens, and A. Torralba, “Sun3d: A database of big spaces reconstructed using sfm and object labels,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1625–1632.

[32] F. Walch, C. H. L. Leal-taix, T. Sattler, and S. H. D. Cremers, “Image-based localization using LSTMs for structured feature correlation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 627–637.

[33] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, “Real-time rgb-d camera relocalization,” in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2013, pp. 173–179.

[34] Y. Li, N. Snavely, and D. P. Huttenlocher, “Location recognition using prioritized feature matching,” in *European conference on computer vision*. Springer, 2010, pp. 791–804.

[35] M. Fallon, H. Johannsson, M. Kaess, and J. J. Leonard, “The mit stata center dataset,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1695–1699, 2013.

[36] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[37] D. Olid, J. M. Fcil, and J. Civera, “Single-view place recognition under seasonal changes,” in *PPNIV Workshop at IROS 2018*, 2018.

[38] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 year, 1000 km: The oxford robotcar dataset,” *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[39] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[40] H. Alismail, B. Browning, and M. B. Dias, “Evaluating pose estimation methods for stereo visual odometry on robots,” in *the 11th Intl Conf. on Intelligent Autonomous Systems (IAS-11)*, vol. 3, 2010, p. 2.

[41] G. Pandey, J. R. McBride, and R. M. Eustice, “Ford campus vision and lidar data set,” *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.

[42] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 573–580.

[43] G. Georgakis, M. A. Reza, A. Mousavian, P.-H. Le, and J. Košecká, “Multiview rgb-d dataset for object instance detection,” in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 426–434.

[44] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.

[45] T. Schops, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, “A multi-view stereo benchmark with

high-resolution images and multi-camera videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3260–3269.

[46] A. Saxena, S. H. Chung, and A. Ng, "Learning Depth from Single Monocular Images," *Advances in Neural Information Processing Systems 18*, pp. 1161–1168, 2006. [Online]. Available: http://books.nips.cc/papers/files/nips18/NIPS2005_0684.pdf

[47] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European conference on computer vision*. Springer, 2012, pp. 611–625.

A.2 Embedded object detection applying Deep Neural Networks in railway domain

This paper was presented in the 23rd Euromicro Conference on Digital System Design (DSD) in 2020, and then published in the conference proceedings. The full citation:

Etxeberria-Garcia, M., Ezaguirre, F., Plazaola, J., Munoz, U., and Zamalloa, M. (2020, August). Embedded object detection applying Deep Neural Networks in railway domain. In *2020 23rd Euromicro Conference on Digital System Design (DSD) (pp. 565-569)*. IEEE.

2020 23rd Euromicro Conference on Digital System Design (DSD)

Embedded object detection applying Deep Neural Networks in railway domain

Mikel Etxeberria-Garcia
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate/Mondragón, Spain
mikel.etxeberria@ikerlan.es

Fernando Ezaguirre
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate/Mondragón, Spain
feizaguirre@ikerlan.es

Joanes Plazaola
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate/Mondragón, Spain
jplazaola@ikerlan.es

Unai Muñoz
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate/Mondragón, Spain
umunoz@ikerlan.es

Maider Zamalloa
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate/Mondragón, Spain
mzamalloa@ikerlan.es

***Abstract*—In the last few years, research on deep learning application on the transportation industry has grown. One of the tasks afforded on those works is the object detection, a essential function in autonomous vehicles, including railway vehicles. While the application of deep learning for object detection is increasing in railway domain, proposed methods have to be yet tested on embedded hardware. This work explores the efficiency of the standard YoloV3 detector embedded on a NVIDIA Jetson AGX Xavier to infer traffic signals in the railway domain. Furthermore, different architectures of YoloV3 are analyzed and compared to find the best output for the used dataset. A data augmentation technique called RICAP-DET is developed to create the training dataset by generating labeled images from cutouts of a set of images. The results show that YoloV3 can be used to detect rail traffic-signals in real time on an embedded platform and that RICAP-DET is adequate to train YoloV3.**

***Index Terms*—deep neural networks, object detection, railway domain, embedded systems, data augmentation**

I. Introduction

Deep Learning evolution of the last decade has been critical for the development of some industrial applications. Since the breakthrough of Convolutional Neural Networks (CNN) for image processing problems, they have been applied for numerous tasks such as classification, localization, or object detection. As in robotics and autonomous vehicles before, the railway domain is also beginning to experiment an increase on deep learning application. Object detection specifically, is essential for many autonomous railway vehicles functions, such as track monitoring or signal detection. In the recent years, deep learning based object detection applications are also emerging in the railway domain. In some of these works, deep learning is used for signal detection [2]–[4] while others use signal detection as a first step for other tasks [5].

Usually, the training of Deep Neural Networks (DNN) occurs on a high-performance computing system, either on-premises or in the cloud. However, previously trained models can be deployed and executed on completely different processing platforms. Railway vehicles (and also track equipment) include some computing systems for control, monitoring and signaling purposes which are embedded in the vehicle itself. These embedded systems are constrained by processing cycles, memory, size or power consumption. Hardware manufactures like NVIDIA are developing new embedded hardware to migrate the application of deep learning approaches towards the industry of autonomous vehicles such as autonomous cars, trains, trams, etc.

In this work we aim to explore the efficiency and limitations of such embedded hardware when affording a deep learning based railway-track signal detection task from the imagines captured by a camera installed in the front of a train.

This paper is divided into the following sections. Section 2 resumes the most relevant previous works in object detection along with the main techniques of data augmentation. All the experimentation phase and data generation are explained in depth in section 3 and finally some conclusions are drawn in section 4.

II. Related work

A. Object detection

In the field of computer vision, different visual recognition problems have been tackled, for instance: image classification,

978-1-7281-9535-3/20/$31.00 ©2020 IEEE
DOI 10.1109/DSD51259.2020.00093

object detection, semantic segmentation or instance segmentation. Object detection locates the presence of an object in an image, performs object classification and localization altogether. In this section the common strategies and architectures used in visual object detection are gathered. All the explored approaches are camera based, as cameras are a non-expensive and non-invasive alternative to other sensors (LIDAR, RADAR, etc.).

In recent years, the detection techniques based on hand-designed features as SIFT [6] or HOG [7] have become obsolete, and many improvement are being achieved in inference speed and detection accuracy by using DNNs. The publication of standard datasets, the cost reduction of storage devices and GPU power allowed the application of deep learning in industry and machine learning practitioners solutions.

Two types of detectors can be found when designing a deep learning based object detector. The used detector type depends on the main objective pursued, which may be the inference speed or the accuracy [17]: one-stage detectors and two-stage detectors.

Recent two-stage detectors predict proposals based on backbone networks, and then an additional classifier is involved for the classification and regression of those proposals. In the two-stage models, the detection is based on regions proposed by selective search or by using a Region Proposal Network. Then, a classifier processes only the candidate regions. Some popular two-stage detectors are R-CNN [8], Fast R-CNN [9], Faster R-CNN [10] or Mask R-CNN [11].

One-stage detectors skip proposal stage and run detection directly over a dense sampling of possible locations. The detection is simpler and faster but might potentially decrease the accuracy. One of the most well-known one-stage object detector was presented by Redmon et al. in 2015: You Only Look Once (YOLO) [13]. YOLO was the first attempt to build a real-time object detector, as it was designed to achieve fast inference speed. It is built with a CNN that is pretrained in ImageNet [16], and the final layer of this network is modified to output a tensor of bounding boxes to localize the objects and the class probabilities.

In 2016, YoloV2 [14] was presented with a series of improvements with respect to the original YOLO, such as, the use of BatchNorm, passthrough layers and the change from using fully connected layers to predict bounding boxes to using CNNs to predict anchor box localization. Later, in 2018, YoloV3 [1] was presented, a detector that uses the DarkNet-53 network as backbone, and the pyramidal feature architecture on top of it, to make predictions at different scales. Moreover, YoloV3 adds residual blocks, skip connections and up sampling, creating a more robust object detector but always maintaining the high inference speed of YOLO.

Outside the YOLO detector family, other approaches have been also proposed over the last few years, such as, SSD [15] or RetinaNet [12].

As this research aims to explore the efficiency of deep learning object detectors in embedded hardware, YoloV3 has been selected for experimentation, as it is one of the most standard and fast (one-stage) object detectors.

B. Data augmentation

One of the main demands of deep learning approaches is the huge data need of the models. Additionally, in the case of supervised learning, another requirement is included where the training data must be labeled. Many methods have arisen to try to reduce the impact of the mentioned data problems, such as, transfer learning, pretraining, one-shot learning or data augmentation. This last method addresses the creation of the training dataset and has been widely used in many deep learning problems. Additionally, some of data augmentation techniques can help generating refined labeled datasets. Most of these methods can be stacked on top of other data augmentation methods.

Basic image manipulation. One of the most basic data augmentation technique is to produce new data transforming the original. Some basic manipulations are based on pixel values as color operations, blurring using kernel filters or edge enhancement. Other augmentations are based on geometric operations, such as, rotation, cropping, translation, scale, flipping, etc. In the case of data augmentation for object detection, most of these techniques can generate new labeled data.

Hinton et al. [19] proposed Dropout augmentation by dropping some pixels on an image and therefore injecting some noise. Another interesting technique is mixing images by averaging their pixel values, also called SamplePairing. In 2018, Takashi et al. proposed Random Image Cropping and Patching (RICAP) [18]. RICAP, randomly crops four images and patches them to construct a new training image. As in the previous case, the generated images are not logic to human eye but are very useful to train DNNs. Additionally, it makes possible to generate accurate labels for objects in object detection task.

Generative Adversarial Networks. Since the fast evolution of deep learning, new strategies have been developed by creating learning data generators. In 2014, Goodfellow et al. invented the idea of Generative Adversarial Network (GAN) [20]. Following this approach, several works appeared where GANs were applied or improved, such as, Coupled GAN (CoGAN) [21], Bidirectional GAN (BiGAN) [22], pix2pix [23] or CycleGAN [24].

Following these works, Antioniu et al. [25] developed Data Augmentation Generative Adversarial Network (DAGAN) to generate new data from sample images. The generation is made using image conditional GANs and these generated images are practically indistinguishable from real images.

Summarizing, several methods of data augmentation are available depending on research interest and existing dataset. However, it is important to consider a search method to find the best augmentation policies as adding too many augmentation methods on top of other augmentation methods not always results on a good training dataset.

Taking all of this into account, we have decided to define a variation of RICAP [18] method, as it is easily implementable, the labelling is done automatically, and it generates enough data to training our model. This variation and how the augmented dataset has been created are explained more in depth in the following section.

III. Experimentation

The aim of this work is to check if a computer vision system embedded in a train could localize in real-time a known type of traffic signals located in the rail track. In this work a simulation has been used recording about 8 minutes at 30 FPS (frames per second) and 1920x1080 resolution. The techniques and experiments described here can be applied to surface trains, tramways, undergrounds, etc. The process of embedding begins with an off-line training of the network in a Workstation and later *exporting* the network into an embedded platform in the train, in this case, an NVIDIA Jetson AGX Xavier [27].

A. Hardware

The Workstation used for training is a Windows 10 PC with an Intel i9-9900k processor with 64GB RAM and an NVIDIA RTX 2080 Ti. The trained networks are tested in an NVIDIA Jetson AGX Xavier Developer Kit that is composed by an ARM 64 bits 8 cores processor, with 32 GB RAM and a GPU Volta with 512 Tensor Cores.

B. Signals detection with Darknet YoloV3

The object detection algorithm used in this work is the Darknet YoloV3 implementation for Windows by AlexeyAB [26]. The *full* YoloV3 variant has been used with 3 and 5 yolo layers. The following sections describe the experimentation main goals:

- Verifying that neural networks trained with the Workstation and later embedded into an NVIDIA Jetson AGX Xavier really do have almost equal performance in terms of detection precision and recall. This will ensure that the process of training and later embedding in the Jetson is working consistently.
- Investigating about the best architectural choices for the neural network for this task. The performance of different architectural alternatives of *Yolo 3 layers* Vs. *Yolo 5 layers* with three different *sizes of input layers* will be measured, comparing how they behave in terms of precision, recall, and in the case of the Jetson, also in terms of FPS. This will result into guidelines and conclusions about the best architectures for the task.
- Experimenting with a variant of data augmentation technique based in cropping.

C. Data Augmentation for Detection

In this paper an innovative cropping type technique inspired in RICAP has been developed. While RICAP is focused on classification, the technique presented in this paper (that we call RICAP-DET) focuses on detection. RICAP-DET was developed to generate a large artificial set of images by composing image cutouts and adding later the signals to be detected. Furthermore, RICAP-DET automates the labeling and annotation of training images using neither manual annotation tools nor semi-automatic video annotation tools (as for example CVAT [30]). The main reason to increase the training dataset was to avoid overfitting during training.

The data augmentation process begins by collecting two groups of images. First, crops with all the types of signals to be detected (as in Figure 1) with different sizes are selected. The size of the images can vary from very small signals of 20x20 pixels up to large signals of 200x200 pixels. In this work, 40 signal images have been selected and they all belong to the same class category (speed signal). Note that the image cutout of the signals must fit only the signal to be detected. Second, a large number of image cutouts representing a variety of railway scenes are selected, ensuring that the selected cutouts do not contain any of the signals to be detected. As an example, Figure 1 shows three image cutouts of railway scenes. In this work, 80 image cutouts of railway scenes have been used.

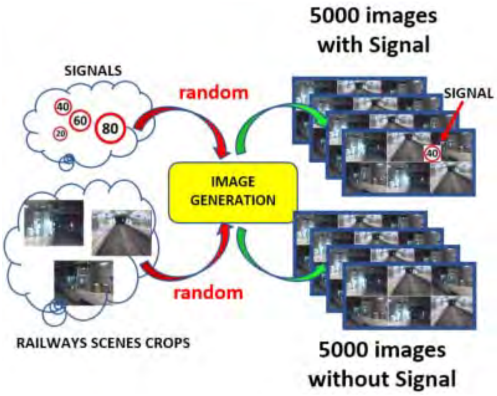


Fig. 1. RICAP-DET data augmentation process

Once a sufficient number of composed images representing a variety of scenes and signals have been collected, the generation process proceeds as follows:

- Generate 10000 images by composing cutouts in a matrix. Each image is composed by randomly selecting 12 cutouts (480x360 pixels) and composing the whole image.
- Randomly select 5000 images out of the 10000 images. For each of the 5000 images, randomly select a signal, draw the signal in a random position on the composed image, and generate the annotation file. Note that the process knows the exact position where the signal is located, thus, can generate the exact annotation.

Some key advantages of this process are that it is possible to generate an arbitrary large number of images to avoid overfitting and the signals to be detected can be *inserted* in scenes in which they were not located originally in the source material, preventing the network to learn that some signals are associated to some specific scenes, thus helping again to avoid overfitting. It is frequent that after training, the network misclassifies objects that are similar to signals.

In this case it is easy to a) select a group of objects that are not signals, b) randomly add the objects to the composed images that are labeled to contain no signal, and c) retrain the network (starting from last weights of the network). This way, the network will learn that those objects are not signals. Furthermore, this procedure can be iterated, obtaining successive versions of the network that learn to exclude the undesired objects. This approach has demonstrated to be very powerful and easily adaptable to new objects, retraining the network in a simple way.

D. Experiment setup

YoloV3 can be configured with multiple parameters and can be trained to detect custom objects by training the final *yolo* layers of the network. In this work, the network has been configured by combining the number of yolo and the size of input layer. In this case, only one type of object will be detected, thus, the network detects only one class: speed signal (shown in Figure 1). Note the different sizes and that the signals can have one or two numbers inside a red circle. In this work, a total of seven YoloV3 configurations have been trained:

- 960 x 960 input layer (with 5 *yolo* layers)
- 608 x 608 input layer (with 5 and 3 *yolo* layers)
- 416 x 416 input layer (with 5 and 3 *yolo* layers)
- 320 x 320 input layer (with 5 and 3 *yolo* layers)

The seven networks share the configuration parameters suggested by YoloV3 authors. Anchors, as required by YoloV3, have been recalculated for each input layer size and each number of *yolo* layers (3 or 5). Additionally, and to speed-up training as recommended by AlexeyAB [26], only the YoloV3 layers are trained. For that purpose, all previous convolution layers weights are preloaded before training.

After training each network in the workstation, it is embedded into the NVIDIA Jetson AGX Xavier. Then, in both platforms, all network variants will be evaluated with the following well known metrics: precision, recall and FPS (measures the throughput of frames that can be achieved).

Once the metrics of all networks combination in both platforms are computed, it must be verified first that the Workstation and the NVIDIA Jetson behaves similarly (in terms of precision and recall) ensuring a consistent embedding process and, second, investigate which architectures show better performance.

E. Results

Regarding the consistency between the workstation and the NVIDIA Jetson Xavier, both platforms perform very similar comparing the precision (Figure 2) and the recall (Figure 3), with negligible differences, therefore ensuring that the embedding process is working properly.

Figure 2 shows the final precision of the seven YoloV3 networks after 4000 epochs in the NVIDIA Jetson Xavier. Precision is shown in Y axis while Intersection Over Union (IoU) threshold is shown in X axis. Figure 3 shows the final general recall of the seven YoloV3 networks after 4000 epochs in the NVIDIA Jetson Xavier. Recall is shown in Y axis while IoU threshold is shown in X axis.

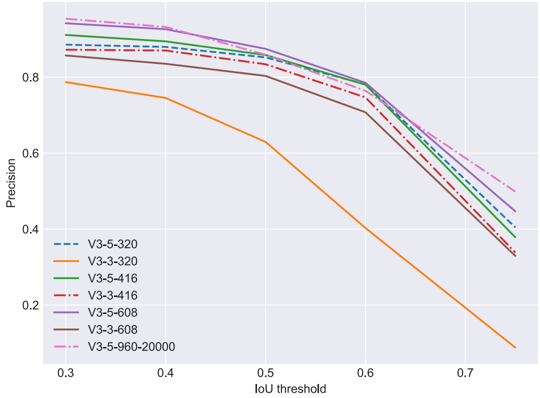


Fig. 2. Precision of the seven networks in the Jetson Xavier

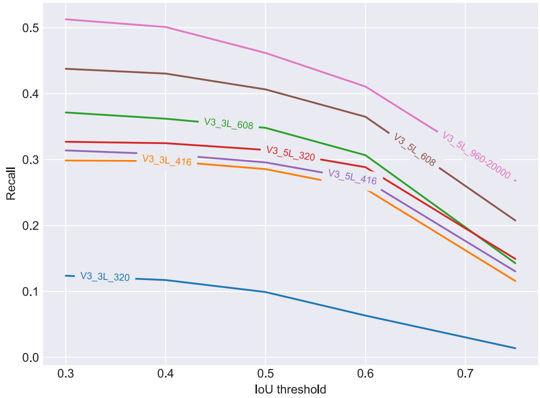


Fig. 3. Recall of the seven networks in the Jetson Xavier

Comparing the precision and the recall of different architectural choices, all the networks (except 3 layers 320x320) performs reasonably well up to a threshold, but degrade quickly from that point on. However, in the case of the recall, there are important differences among them, the best being the one with 5 layers and input size of 960x960. In general terms, it is clear that the network performs better with greater amount of layers and larger input sizes. One interesting result emerges from the transition between 5 and 3 layer: *the size of the input layer can have a larger impact in the precision and recall than having more layers*. This is demonstrated by the fact that the 3 layer 608x608 solution is better at recall than the 5 layer 416x416 and the 5 layer 320x320 solution. Therefore, the size of the input layer of the network is important: the degree of scaling that YoloV3 must apply is a key factor.

Figure 4 shows the FPS (frames per second) achieved by the NVIDIA Jetson AGX Xavier with 32-bit and 16-bit float precision. Note that while precision and recall are similar in 32-bit and 16-bit version (Figures 2 and 3), there is however a huge difference in terms of FPS in favor of the 16-bit versions. Note also that in the transition (5 layer 16-bit and 3 layer 32-bit) the FPS are very similar.

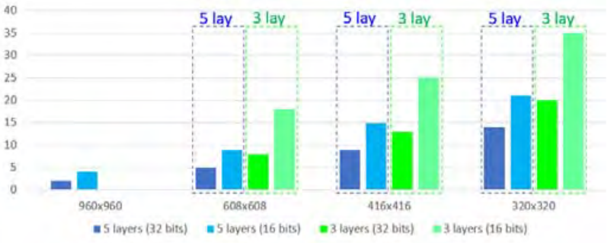


Fig. 4. FPS of the seven networks in the Jetson Xavier with 32-bit and 16-bit float precision

IV. Conclusion

This paper analyzes the possibility of embedding a YoloV3 network into an NVIDIA Jetson Xavier platform in the railway domain for real-time detection of traffic signals located along the track. Several possible configurations have been trained in a workstation and tested on an embedded platform. To make a wise choice in a real life applications it is always recommended to study several types of architectural choices in terms of number of Yolo *layers*, size of Yolo *input* layer and 32-bit or 16-bit precision.

From results, it can be verified that the networks behave similarly in both platforms (workstation and Jetson). It has been found that it is feasible to use an NVIDIA Jetson AGX Xavier embedded in a train to detect traffic signals in real-time. The best architectural choice, from all the evaluated ones, is the 3 Yolo *layers* with a 608x608 input layer and 16-bit float precision, achieving a throughput of 18 FPS.

Regarding the used data augmentation technique called RICAP-DET, it has shown good results and has proved being a viable choice to create a dataset to train a detection network on a semi-automatic way.

Since the experimentation of this research concluded, a new promising approach called EfficientNet has been presented by Tan et al. [31]. In this work, each of the detector component is improved and the model obtains remarkable results, improving by far the current state-of-the-art results, hence, its future application is foreseen.

References

[1] Redmon, J., and Farhadi, A., "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767. 2018.

[2] Ritika, S., Mittal, S., and Rao, D., "Railway track specific traffic signal selection using deep learning." arXiv preprint arXiv:1712.06107 (2017).

[3] Choodowicz, E., Lisiecki, P., and Lech, P., "Hybrid Algorithm for the Detection and Recognition of Railway Signs." International Conference on Computer Recognition Systems. Springer, Cham, 2019.

[4] Karagiannis, G., Olsen, S., and Pedersen, K., "Deep Learning for Detection of Railway Signs and Signals." Science and Information Conference. Springer, Cham, 2019.

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

[6] Lowe, D. G., "Distinctive image features from scale-invariant keypoints." International journal of computer vision, 60(2), 91-110. 2004.

[7] Dalal, N., and Triggs, B., "Histograms of oriented gradients for human detection." 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Vol. 1. IEEE, 2005.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.

[9] Girshick, R., "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.

[10] Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

[11] He, K., Gkioxari, G., Dollár, P., and Girshick, R., "Mask r-cnn." Proceedings of the IEEE international conference on computer vision. 2017.

[12] Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P., "Focal loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017.

[13] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[14] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517–6525. IEEE, 2017.

[15] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., and Berg, A. C. "SSD: Single shot multibox detector." European conference on computer vision. Springer, Cham, 2016.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR, 2009.

[17] Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., and Qu, R., "A Survey of Deep Learning-Based Object Detection." IEEE Access 7 (2019): 128837-128868.

[18] Takahashi, R., Matsubara, T., and Uehara, K., "RICAP: Random image cropping and patching data augmentation for deep CNNs." Asian Conference on Machine Learning. 2018

[19] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R., "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580. 2012.

[20] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... and Bengio, Y., "Generative adversarial nets." In Advances in neural information processing systems (pp. 2672-2680). 2014.

[21] Liu, M. Y., and Tuzel, O. "Coupled generative adversarial networks." In Advances in neural information processing systems (pp. 469-477). 2016.

[22] Donahue, J., Krähenbühl, P., and Darrell, T., "Adversarial feature learning." arXiv preprint arXiv:1605.09782. 2016.

[23] Isola, P., Zhu, J. Y., Zhou, T., and Efros, A. A., "Image-to-image translation with conditional adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134). 2017.

[24] Zhu, J. Y., Park, T., Isola, P., and Efros, A. A., "Unpaired image-to-image translation using cycle-consistent adversarial networks." In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232). 2017.

[25] Antoniou, A., Storkey, A., and Edwards, H., "Data augmentation generative adversarial networks." arXiv preprint arXiv:1711.04340. 2017.

[26] AlexeyAB, Darknet YoloV3 implementation. https://github.com/AlexeyAB/darknet.

[27] NVIDIA Jetson AGX Xavier, https://www.nvidia.com/es-es/autonomous-machines/embedded-systems/jetson-agx-xavier/.

[28] NVIDIA TensorRT 6, https://developer.nvidia.com/tensorrt. 2019.

[29] Connor Shorten, Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, (2019) 6:60. https://doi.org/10.1186/s40537-019-0197-0.

[30] CVAT, https://github.com/opencv/cvat.

[31] Tan, M., and Le, Q. V., "Efficientnet: Rethinking model scaling for convolutional neural networks." arXiv preprint arXiv:1905.11946 (2019).

A.3 Monocular visual odometry for underground railway scenarios

This paper was presented in the IEEE/SICE Fifteenth International Conference on Quality Control by Artificial Vision (QCAV2021) in 2021, and then published in the SPIE proceedings 11794. The full citation:

Etxeberria-Garcia, M., Labayen, M., Eizaguirre, F., Zamalloa, M., and Arana-Arexolaleiba, N. (2021, July). Monocular visual odometry for underground railway scenarios. In *Fifteenth International Conference on Quality Control by Artificial Vision (Vol. 11794, p. 1179402)*. International Society for Optics and Photonics.

Monocular visual odometry for underground railway scenarios

Etxeberria-Garcia, Mikel, Labayen, Mikel, Eizaguirre, Fernando, Zamalloa, Maider, Arana-Arexolaleiba, Nestor

Mikel Etxeberria-Garcia, Mikel Labayen, Fernando Eizaguirre, Maider Zamalloa, Nestor Arana-Arexolaleiba, "Monocular visual odometry for underground railway scenarios," Proc. SPIE 11794, Fifteenth International Conference on Quality Control by Artificial Vision, 1179402 (16 July 2021); doi: 10.1117/12.2586310

Event: Fifteenth International Conference on Quality Control by Artificial Vision, 2021, Tokushima, Japan

Monocular Visual Odometry for underground railway scenarios

Mikel Etxeberria-Garcia[a], Mikel Labayen[b], Fernando Eizaguirre[a], Maider Zamalloa[a], and Nestor Arana-Arexolaleiba[c]

[a]Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA). P.º J.M. Arizmendiarrieta, 2. 20500 Arrasate/Mondragón, Gipuzkoa, Spain
[b]Faculty of Informatics, UPV/EHU, Manuel Lardizabal Ibilbidea, 1, 20018 Donostia, Gipuzkoa, Spain
[c]MGEP, Mondragon Unibertsitatea, Loramendi Kalea, 4 20500 Arrasate/Mondragon Gipuzkoa, Spain

ABSTRACT

In this paper, the application of monocular Visual Odometry (VO) solutions for underground train stopping operation are explored. In order to analyze if the application of monocular VO solutions in challenging environments as underground railway scenarios is viable, different VO architectures are selected. For that, the state of the art of deep learning based VO approaches is analyzed. Four categories can be defined in the VO approaches defined in the last few years: (1) supervised pure deep learning based solutions; (2) solutions combining geometric features and deep learning; (3) solutions combining inertial sensors and deep learning; and (4) unsupervised deep learning solutions. A dataset composed of underground train stop operations was also created, where the ground truth is labeled according to the onboard unit SIL-4 ERTMS/ETCS odometry data. The dataset was recorded by using a camera installed in front of the train. Preliminary experimental results demonstrate that deep learning based VO solutions are applicable in underground train stop operations.

Keywords: computer vision, rail transportation, deep learning, artificial intelligence, autonomous train

1. INTRODUCTION

The evolution of technology in the field of Computer Sciences and Artificial Intelligence (AI) is transforming the world around us with revolutionary solutions such as autonomous vehicles, medicine or 4.0 industry. These advances are allowing the introduction of some technological solutions in the society that a few years ago was only a visionary idea: virtual assistants and autonomous vehicles within others. In this context, the development of Computer Vision (CV) and AI are opening new opportunities that allow migrating towards an autonomous operation of vehicles.

The railway domain is transforming towards the *Intelligent Transportation Systems* (ITS) and the *Advanced Driving Assistance Systems* (ADAS) industry. The Communication-Based Train Control (CBTC) standard (IEEE 1474[1]) defines an automatic train control system where the track and onboard subsystems are continuously communicated. In an autonomous train system all the operations involved must be automatic, for example, operations such as train doors opening/closing, gauge control in platforms, or train stopping or coupling. In this context, accurate train localization and platform-train doors alignment are required for a safe passenger transfer train operation.

Nowadays, some of the technologies that estimate the train position are based on wheel odometry and radars: a beacon-based system in the track and, encoders and radars installed onboard to estimate train odometry data. The inaccuracy of radar and encoder sensors estimation is corrected when the onboard controlling system receives track beacon distance information.

Further author information: send correspondence to Mikel Etxeberria-Garcia (mikel.etxeberria@ikerlan.es).

Fifteenth International Conference on Quality Control by Artificial Vision, edited by Kenji Terada
Akio Nakamura, Takashi Komuro, Tsuyoshi Shimizu, Proc. of SPIE Vol. 11794, 1179402
© 2021 SPIE · CCC code: 0277-786X/21/$21 · doi: 10.1117/12.2586310

However, at a stopping point, the driver's eyes and experience are still the key factors to align the train correctly with the platform area and to remove the final localization error. Furthermore, the beacon-based system has a high cost, as a lot of beacons must be placed in the rail infrastructure, maintenance cost is high, and the deployment is slowed down.[2] Additionally, some researches state that sensing infrastructure may be replaced with cheaper sensors leading to a more cost-effective solution (Tschopp *et al.*[3]), for example, cameras. These visual sensors are a low-cost technology, have availability, and have been successfully used for localization task in multiple domains. The motivation of this research comes from the need of overcoming to the drawbacks of current perception systems regarding autonomous train stopping operation.

Recent advances in Visual Odometry, the branch of computer vision that is responsible for the pose estimation and localization of autonomous systems using motion changes in camera images, have arisen new approaches. Furthermore, deep learning advances have enhanced these new approaches. However these approaches are usually tested on known concrete environments. **This research aims to explore the application of monocular VO algorithms for train localization in underground scenarios**.

This paper is divided into the following sections. In Section 2, Visual Odometry (VO) approaches of the literature are explored followed by the data generation process in Section 3. Then, the applications of the selected algorithms in the railway domain are presented in Section 4 and, finally, conclusions are drawn in Section 5.

2. RELATED WORK

The use of Machine Learning (ML) approaches in Computer Vision problems has grown with the increase of computational resources and new Deep Learning (DL) architectures. This situation also comes from the promising results obtained by the application of DL methods in CV, specifically with the use of Convolutional Neural Networks (CNN) on large-scale image classification (Krizhevsky *et al.*[4]). Additionally, it has been shown that one of the potentials of Deep Neural Networks (DNN) is their generalization ability in visual recognition tasks.

Focusing on the railway domain, some Deep Learning based approaches can be found, but not as much as in other common fields as Computer Vision and robotics. Some of the works focus on rail track inspection or monitoring.[5,6] Other researches are more focused on risk assessment.[7] Object (signal) detection is also a main task for railway domain researches as in[8] or in.[9] In this last work, Haseeb *et al.* defined DisNet, a method to estimate the distance to the objects detected by a monocular camera installed on a train. It makes a regression based on the size of the detected objects using YOLO.[10] Later in 2019, Multi-DisNet was presented in[11] where previous work was improved using multiple cameras.

Concerning VO applications in the railway domain, Tschopp *et al.* perform an experimentation of VO Methods on Rail Vehicles.[3] They study geometric outdoor VO methods and introduce some Visual-Inertial Odometry algorithms by adding stereo cameras and an inertial sensor to their setup. Apart from this work, there is not much research on VO application in the railway domain. Therefore, approaches from other domains have to be considered, and their applicability has to be analyzed. Following this, deep learning based approaches can solve some of the geometric monocular VO approaches problems as their predictions are associated with a real-world scale.

Visual Odometry (VO). Visual Odometry (VO) is a particular case of odometry where the position information is acquired through camera images. The term Visual Odometry was first introduced by Niester *et al.*[12] VO techniques can be classified as geometric-based or learning-based. Even the geometric approaches have received a lot of attention, still in 2017, ORB-SLAM2[13] was presented, an efficient and accurate geometric localization approach that has become the state-of-the-art comparison for all the later approaches. It is based on feature matching consecutive frames and on bundle adjustment algorithm. Geometric based approaches show particularly good when there are enough illumination and texture to match features among different frames, sufficient overlap between frames and scene is static.[3] However, geometric VO works suffer from scale drift issue where scale is inconsistent and expensive global bundle adjustment algorithms are applied to minimize the problem. Furthermore, monocular VO algorithms have a depth-translation scale ambiguity issue.

DL-based approaches can solve the scale-ambiguity issue in monocular VO as their predictions are associated with a real-world scale. In recent years, there have been several works based on monocular VO and DL. They can be classified in the following four categories (see Figure 1): (1) Supervised Deep Learning VO solutions; (2) VO solutions that combine learning based and geometric techniques/features; (3) Solutions that include an IMU in deep learning based VO approaches, and (4) Unsupervised Deep Learning VO solutions.

1. **Supervised Deep Learning VO.** In 2017, DeepVO[14] was created which infers camera poses directly on an end-to-end manner from a sequence of RGB frames. It can produce accurate results and has a good generalization ability as it works well in new scenarios. Recently, D3VO[15] was presented which infers camera pose, depth and uncertainty altogether. The authors claims that it out-performs state-of-the-art stereo/Lidar based methods. However, it is not a fully monocular work as the network is trained using stereo images.

2. **Deep Learning + geometric.** Semi-direct visual odometry (SVO)[16] is the state-of-art of feature matching methods that combines direct and indirect methods offering an efficient probabilistic mapping method. Based on SVO, CNN-SVO[17] introduced an improved using a depth prediction neural network when initializing the map point. It was the first work mixing geometric and learning-based odometry methods. Recently, Zhan *et al.* presented DF-VO.[18] It outperforms pure deep learning-based and geometry-based methods and solves the scale-drift issue by adding a scale consistent single-view depth CNN. The training can be done using monocular or stereo datasets.

3. **Deep Learning + IMU.** Lately, some approaches have introduced IMU sensor to improve pure deep learning-based methods. Han *et al.*[19] presented DeepVIO, a method that merges mono-camera optical flow and IMU trough three neural networks. DeepVIO shows state of the art results in terms of accuracy and data adaptability while reducing the impact of inaccurate calibrations and unsynchronized or missing data. To avoid the need for ground-truth labelling, they use a self-supervised learning framework, as they do the next articles.

4. **Unsupervised Deep Learning.** In 2018, a new unsupervised learning framework called SFMLearner was proposed by Kathpal *et al.*[20] It infers depth and camera motion using only monocular video sequences to train the network based on scene structure and view synthesis. However, assumes camera intrinsic are given and does not explicitly estimate scene dynamics. In 2019, SC-SFMLearner[21] was proposed that improves SFMLearner with a geometric consistency loss to solve the scale ambiguity over the frames and with a self-discovered mask to handle moving objects and occlusions. SC-SFML can estimate scale-consistent camera trajectory over a long sequence.

   DVSO[22] leverages deep monocular depth prediction to overcome limitations of geometry-based monocular VO. It is a semi-supervised approach, trained with stereo images. It achieves state-of-art accuracy for monocular and deep learning-based visual odometry while in performance it can be compared with stereo methods. It recovers metric scale and reduces scale drift in geometric monocular VO. The generalization ability of DVSO must be analyzed yet. In 2018, Depth-VO-Feat[23] was proposed by Zhan *et al.* Depth-VO-Feat is an unsupervised monocular VO framework trained with stereo video sequences for learning depth and VO. They propose a novel feature reconstruction loss without scale ambiguity. However, no occlusion assumption is made, and the scene must be rigid. Furthermore, the training is done using stereo video sequences while we focus on training with a monocular dataset. In 2019, Shen *et al.*[24] presented DeepMatchVO, a self-supervised monocular approach for VO. They introduce the matching loss that includes the photometric loss and the geometric loss to avoid significant systematic errors due to occlusions and reflective surfaces. The authors claim that it outperforms previous unsupervised monocular approaches.

However, none of the previous works have dealt with challenging scenarios as underground railway. These scenarios are characterized by some conditions that can hinder the application of state-of-the-art VO algorithms. As Almalioglu *et al.*[25] point out, these techniques usually rely on finding correspondences between consecutive frames, which can not be made accurately in challenging environments as they infringe fundamental conditions such as stable lightning, lambertian surfaces and variable textures. Underground railway scenarios can be included into those challenging environments as they include non-lambertian surfaces or big lighting changes, being low in the tunnel, and high in the stations where part of the images are saturated. Besides, cameras tuned to work in both lighting conditions, give blurring and noisy images.

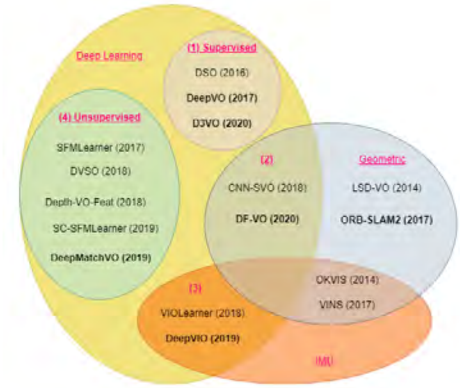


Figure 1. State-of-the-art works in Visual Odometry approaches classified into four categories: (1) supervised pure deep learning based solutions; (2) solutions combining geometric features and deep learning; (3) solutions combining inertial sensors and deep learning; and (4) unsupervised deep learning solutions.

Taking into account the previous related work classification and the issues of challenging scenarios, in this paper we propose to compare the application of two different approaches: geometric state-of-the-art approach **ORB-SLAM2** and DL+geometric hybrid solution **DF-VO**. Furthermore, DF-VO algorithm uses a depth model that can be trained to evaluate if the results may be improved.

3. EXPERIMENTATION

Taking into account the previous related work categorization and the generated dataset, an experimental and theoretical study of two monocular VO algorithms is being made: geometry based state-of-the-art approach **ORB-SLAM2** and DL+geometric hybrid solution **DF-VO**. The importance of having a dataset with a precise ground truth is well known in the DL community. This ground truth can be used to train supervised networks or to evaluate unsupervised networks. Usually, Visual Odometry researches are evaluated using the standard KITTI dataset[26] (specifically sequences 09-10), but KITTI does not include underground railway environment data. Consequently, one of the tasks to afford was the collection of an underground railway scenario dataset.

3.1 Dataset generation

The data for DL algorithms can be collected from different sources: from simulated environments, from existing datasets, and datasets explicitly recorded and labeled in a railway scenario. From an analysis of the most used dataset in Deep Learning applications for autonomous navigation,[27] no existing dataset was found that fit the requirements stated in this research. A unique database of the railway domain was identified (Norland[28]) but it is recorded in an outdoor environment and the train does not stop on stations. Therefore, it is out of underground train stop operation scenario use-case.

Simulated environments offer powerful possibilities to generate data when no real railway applications are involved, because the dataset can be created automatically. However, simulators do not necessarily replicate real-world scenario conditions accurately to validate industrial applications. This situation raised the need to collect and label a new dataset.

The recording was done by a camera installed in a train running in the railway called *Line 3 - Bilbao*. This train covers five underground stations, with a total length of 5,8 km, and each station is recorded more than once in different directions. A monocular camera was installed in front of the train to capture the data. The captured data is synchronized with onboard unit odometry data extracted from replicated radar and encoder sensors and used as reference. The synchronization was done using a self-made application for frame processing and on an onboard monitoring system to capture the data. The synchronized data includes speed and traveled distance, and also other derived information as the distance to the stopping point, and distance from the last beacon.

Station	Sequence ID	Number of frames	Sequence length (m)
Txurdinaga	00	946	64.14
	01	504	65.62
	02	462	61.33
Otxarkoaga	03	719	68.35
	04	546	64.15
	05	462	67.16
	06	515	69.48
Uribarri	07	529	66.85
	08	488	62.54
Zurbaranbarri	09	549	67.02
	10	594	68.60
	11	500	61.31
Zazpikaleak	12	504	63.82

Table 1. Dataset generated from railway Line 3-Bilbao. The table resumes the sequence number, the number of frames from each sequence and the ground truth length of each sequence.

The accuracy of distance estimation is directly related to scenes' characteristic points sharpness and the accuracy in their detection. For this reason, blur effect reduction is essential in image capture process limiting the shutter time. Unfortunately, one of the undesirable effects derived from this action is obtaining low-light images. This effect is corrected adding electronic gain in the camera, but this action in turn adds noise to images. In order to get balanced solutions and considering that the maximum train speed in this research is 90km/h, shutter time is limited to a maximum of 12ms, electronic gain to a maximum of 25dB and lens aperture is set to F1.2. Fig. 2 shows one of the frames from the collected dataset. The used camera is a *Basler acA1920-40uc* with a sensor *Sony IMX249 color CMOS* and a lens *Fujinon DF6HA-1S* at a frame rate of 25 fps (50 Hz). The full generated dataset is depicted in table 1.

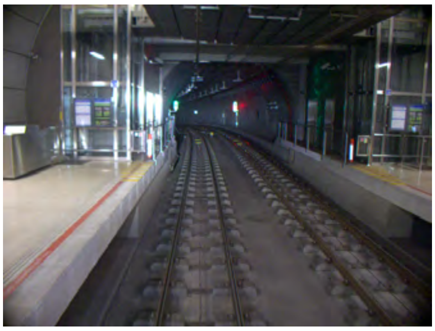


Figure 2. Sample image of the generated dataset from a real underground railway scenario. This dataset contains 13 sequences of a train stopping on various stations from line 3 in Bilbao.

3.2 VO evaluation

The applicability of the algorithms is evaluated using the ego-motion from odometry obtained from the train. As all the sequences from the railway environment are focused on the platforms and barely rotate (platforms are straight), just the translation is taken into account. The comparison is made using three standard metrics. First, *Absolute Trajectory Error (ATE)* is used where the root-mean-square error between estimated trajectory poses and the respective reference are measured. For relative evaluation, *Relative Pose Error (RPE)* and *average translational error* (*t_{err}*) are common metrics on VO evaluation. The RPE measures the local accuracy of the trajectory per frame.[29] Therefore, the relative pose error corresponds to the drift of the trajectory which is in particular useful for the evaluation of VO systems. Following the underground challenging environments' evaluation criteria proposed by Almalioglu *et al.*[25] , average translational error $t_{err}(\%)$ is used on sub-sequences of different lengths at different speeds (the speeds vary in different sequences as the train is stopping). It is calculated in the same way as RPE, but instead of calculating the relative error on the full sequence, the error on some sub-sequences is calculated and the averaged.

To calculate ATE, first Horn[30] method is used to align both trajectories and find the rigid-body transformation S between them. Given this transformation, the trajectory error matrix at time i as

$$E_i^{ATE} := Q_i^{-1} S P_i \tag{1}$$

where P_i and Q_i are the estimated pose and reference pose at time i respectively. For relative evaluation relative error matrix at time i is calculated as

$$E_i^{RPE} := \left(Q_i^{-1} Q_{i+\Delta}\right)^{-1} \left(P_i^{-1} P_{i+\Delta}\right) \tag{2}$$

where usually $\Delta = 1$ meaning that two consecutive poses are used. Then the root-mean-square error (RMSE) from error matrices is calculated to obtain the ATE and the RPE:

$$RMSE = \left(\frac{1}{m} \sum_{i=1}^{m} \|trans(E_i)\|^2\right)^{\frac{1}{2}} \tag{3}$$

where m is the quantity of error matrices from a sequence of n poses which is $m = n$ in absolute metrics and $m = n - \Delta$ in relative metrics. As stated before, the average translational error t_err is calculated in the same way as RPE, but instead of calculating the relative error from frame to frame over all the full sequence, the average error in sub-sequences of length $\{7, 14, 21, 28, 35, 42, 49, 56\}$ m is used. The relative error matrix is calculated as in 2. Then, RMSE over all sub-sequences RPE is calculated.

Dataset	Metric	00	01	02	03	04	05	06	07	08	09	10	11	12	Avg. Err.
DF-VO (KITTI)	t_{err} (%)	12.15	5.52	10.7	9.22	7.05	6.89	5.94	5.77	11.46	9.04	10.71	12.76	3.10	8.49
	ATE (m)	1.64	1.04	1.54	1.46	1.12	1.21	1.14	0.94	1.78	1.58	1.79	2.06	0.56	1.37
	RPE (m)	0.014	0.016	0.024	0.014	0.026	0.046	0.015	0.012	0.027	0.019	0.020	0.024	0.009	0.020
DF-VO (Line 3-Bilbao)	t_{err} (%)	10.44	6.29	6.34	9.95	8.78	8.9	6.92	10.09	8.34	7.24	10.17	6.16	4.66	8.02
	ATE (m)	1.07	0.58	0.6	1.47	1.32	1.51	1.15	1.52	1.24	1.22	1.83	0.95	0.64	1.162
	RPE (m)	0.023	0.031	0.031	0.019	0.031	0.049	0.022	0.019	0.029	0.031	0.032	0.029	0.014	0.028

Table 2. Comparison of DF-VO application were depth estimation model from Monodepth2 is trained on KITTI dataset and on Line 3-Bilbao using monocular model 640x192.

3.3 VO application

As stated before, two VO algorithms have been selected in this research. ORB-SLAM2[13] is a well-known accurate geometry based VO approach that has been used as state-of-the-art comparison for DL based VO algorithms. It relies on ORB features and then it performs frame tracking, local mapping and loop closing steps to reconstruct the environment while estimate the camera trajectory. When using a monocular camera, geometry based algorithms suffer from some issues. As depth is not observable from just one camera, there is a scale issue that affects the estimation of the map and the trajectory. In addition, the triangulation of the initial map depends on multi-view or filtering techniques and reconstruction may fail when performing pure rotations.

From experimentation is observed that ORB-SLAM2 does not work in Line 3-Bilbao dataset. This makes sense as the tested environment contains some characteristics that violate the conditions of image correspondence between consecutive frames (big light changes and nono-lambertian surfaces). However DL-based algorithm DF-VO[18] overcomes those limitations. DF-VO is an unsupervised DL+geometric hybrid VO algorithm that is based on depth and flow estimation. These estimations are made using the unsupervised algorithm Monodepth2[31] for depth and LiteFlowNet[32] for flow estimation. Monodepth2 learns depth from a stereo pair (consecutive frames from monocular camera) minimizing the reprojection error and auto-masking stationary and occluded pixels.

The results show that DF-VO is capable of obtaining a mean average translational error of 8.02 in a challenging environmet as Line 3-Bilbao, which is a similar results to other results obtained on the standard KITTI odometry dataset by other VO methods such as SfM-Learner[33] (14.068), Depth-VO-Feat[23] (7.911), ORB-SLAM2[13] (8.074) or CNN-SVO[17] (12.663).

Furthermore, DF-VO has been evaluated with two depth models to asses if the training process improves the results in these challenging environments: monocular model 640x192 trained in KITTI dataset and monocular model 640x192 retrained in Line3-Bilbao dataset. As it is shown in table 2, the training process improves the results.

4. CONCLUSIONS

In this paper, the application of monocular VO solutions in underground railway scenarios for train stop operation are explored. Geometric VO approaches are limited by some key factors like illumination, textures, subsequent frames overlap or whether the scene is static or not. In this study is analyzed if deep learning based VO approaches can afford some of the limitations of geometric VO solutions. For that, a geometric approach and a hybrid Deep Learning approach were chosen to explore its applicability in underground railway scenarios. To test the applicability of the selected approach, an underground railway scenario dataset was created by using a monocular camera installed in front of the train.

Geometric approach ORB-SLAM2 seems to fail on underground railway platforms as some of the minimum geometric requirements are not full-filled on this challenging environment: platform surfaces and big lightning changes (saturation). However, when other type of environments are selected, such as underground railway tunnels, geometric ORB-SLAM2 does not fail due to scenes attributes. Although hybrid approaches as DF-VO are also based on geometric features, estimating the flow and the depth from deep learning algorithms makes DF-VO less dependent to scenario characteristics and this algorithm obtains comparable results in this environment. Furthermore, training the depth network under DF-VO for this environment improves the results. Nonetheless, the algorithm suffers from underground environment conditions in other scenarios outside railway platforms.

These results could be boosted by adapting the algorithms to the characteristics of these challenging environments so it can handle the changing lightning conditions or the non-lambertian surfaces. This study also suggests to analyze if other deep learning VO techniques that have shown good results in robotics are also adequate for underground railway scenarios (i.e. unsupervised deep learning solutions, approaches that combine inertial sensoring information with deep learning, solutions that include LiDAR). Therefore, future work includes the creation of an underground railway scenario dataset recorded by a stereo camera, IMU sensors or LiDAR and increasing the current monocular dataset.

ACKNOWLEDGMENTS

This work was supported in part by the Basque Government through BIKAINTEK2018 program and CAF Signalling. In collaboration between Ikerlan, CAF Signalling and Mondragon Unibertsitatea.

REFERENCES

[1] "Ieee standard for communications-based train control (cbtc) performance and functional requirements," *IEEE Std 1474.1-2004 (Revision of IEEE Std 1474.1-1999)* , 1–45 (2004).

[2] Marais, J., Beugin, J., and Berbineau, M., "A Survey of GNSS-Based Research and Developments for the European Railway Signalling," *IEEE Transactions on Intelligent Transportation Systems* **18**(10), 2602–2618 (2017).

[3] Tschopp, F., Schneider, T., Palmer, A. W., Nourani-Vatani, N., Cadena Lerma, C., Siegwart, R., and Nieto, J., "Experimental Comparison of Visual-Aided Odometry Methods for Rail Vehicles," *IEEE Robotics and Automation Letters* , 1–1 (2019).

[4] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105 (2012).

[5] Gibert, X., Patel, V. M., and Chellappa, R., "Deep multitask learning for railway track inspection," *IEEE transactions on intelligent transportation systems* **18**(1), 153–164 (2016).

[6] Mittal, S. and Rao, D., "Vision based railway track monitoring using deep learning," *arXiv preprint arXiv:1711.06423* (2017).

[7] "A deep learning approach towards railway safety risk assessment," *IEEE Access* **PP**, 1–1 (05 2020).

[8] Karagiannis, G., Olsen, S., and Pedersen, K., "Deep learning for detection of railway signs and signals," in [*Science and Information Conference*], 1–15, Springer (2019).

[9] Haseeb, M. A., Guan, J., Ristić-Durrant, D., and Gräser, A., "Disnet: A novel method for distance estimation from monocular camera," (2012).

[10] Redmon, J. and Farhadi, A., "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767* (2018).

[11] Abdul, H. M., Danijela, R.-D., Axel, G., Milan, B., and Dušan, S., "Multi-disnet: Machine learning-based object distance estimation from multiple cameras," in [*International Conference on Computer Vision Systems*], 457–469, Springer (2019).

[12] Nistér, D., Naroditsky, O., and Bergen, J., "Visual Odometry," in [*Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*], 1 (2004).

[13] Mur-Artal, R. and Tardós, J. D., "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics* **33**(5), 1255–1262 (2017).

[14] Wang, S., Clark, R., Wen, H., and Trigoni, N., "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in [*2017 IEEE International Conference on Robotics and Automation (ICRA)*], 2043–2050, IEEE (2017).

[15] Yang, N., von Stumberg, L., Wang, R., and Cremers, D., "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry," *arXiv preprint arXiv:2003.01060* (2020).

[16] Forster, C., Pizzoli, M., and Scaramuzza, D., "Svo: Fast semi-direct monocular visual odometry," in [*2014 IEEE international conference on robotics and automation (ICRA)*], 15–22, IEEE (2014).

[17] Loo, S. Y., Amiri, A. J., Mashohor, S., Tang, S. H., and Zhang, H., "Cnn-svo: Improving the mapping in semi-direct visual odometry using single-image depth prediction," in [*2019 International Conference on Robotics and Automation (ICRA)*], 5218–5223, IEEE (2019).

[18] Zhan, H., Weerasekera, C. S., Bian, J., and Reid, I., "Visual odometry revisited: What should be learnt?," *arXiv preprint arXiv:1909.09803* (2019).

[19] Han, L., Lin, Y., Du, G., and Lian, S., "Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints," *arXiv preprint arXiv:1906.11435* (2019).

[20] Kathpal, A., Shah, D., and Pathak, M., "Learning the structure from motion, an unsupervised approach," (2018).

[21] Bian, J., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.-M., and Reid, I., "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in [*Advances in Neural Information Processing Systems*], 35–45 (2019).

[22] Yang, N., Wang, R., Stuckler, J., and Cremers, D., "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in [*Proceedings of the European Conference on Computer Vision (ECCV)*], 817–833 (2018).

[23] Zhan, H., Garg, R., Saroj Weerasekera, C., Li, K., Agarwal, H., and Reid, I., "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 340–349 (2018).

[24] Shen, T., Luo, Z., Zhou, L., Deng, H., Zhang, R., Fang, T., and Quan, L., "Beyond photometric loss for self-supervised ego-motion estimation," in [*2019 International Conference on Robotics and Automation (ICRA)*], 6359–6365, IEEE (2019).

[25] Almalioglu, Y., Santamaria-Navarro, A., Morrell, B., and Agha-mohammadi, A.-a., "Unsupervised deep persistent monocular visual odometry and depth estimation in extreme environments," *arXiv preprint arXiv:2011.00341* (2020).

[26] Geiger, A., Lenz, P., and Urtasun, R., "Are we ready for autonomous driving? the kitti vision benchmark suite," in [*Conference on Computer Vision and Pattern Recognition (CVPR)*], (2012).

[27] Etxeberria-Garcia, M., Labayen, M., Zamalloa, M., and Arana-Arexolaleiba, N., "Application of computer vision and deep learning in the railway domain for autonomous train stop operation," in [*2020 IEEE/SICE International Symposium on System Integration (SII)*], 943–948, IEEE (2020).

[28] Olid, D., Fácil, J. M., and Civera, J., "Single-view place recognition under seasonal changes," in [*PPNIV Workshop at IROS 2018*], (2018).

[29] Sturm, J., Burgard, W., and Cremers, D., "Evaluating egomotion and structure-from-motion approaches using the tum rgb-d benchmark," in [*Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS)*], (2012).

[30] Horn, B. K., "Closed-form solution of absolute orientation using unit quaternions," *Josa a* **4**(4), 629–642 (1987).

[31] Godard, C., Mac Aodha, O., Firman, M., and Brostow, G. J., "Digging into self-supervised monocular depth prediction," (October 2019).

[32] Hui, T.-W., Tang, X., and Loy, C. C., "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation," in [*Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 8981–8989 (2018).

[33] Zhou, T., Brown, M., Snavely, N., and Lowe, D. G., "Unsupervised learning of depth and ego-motion from video," in [*Proceedings of the IEEE conference on computer vision and pattern recognition*], 1851–1858 (2017).

A.4 Image Enhancement using GANs for Monocular Visual Odometry

This paper was presented in the IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM) in 2021, and then published in proceedings. The full citation:

Ansorregi, J. Z., Garcia, M. E., Akizu, M. Z., and Arexolaleiba, N. A. (2021, June). Image Enhancement using GANs for Monocular Visual Odometry. *In 2021 IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM) (pp. 1-6).* IEEE.

2021 IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM) | 978-1-5386-1757-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ECMSM51310.2021.9468831

Image Enhancement using GANs for Monocular Visual Odometry

Jon Zubieta Ansorregi
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate/Mondragón, Spain.
jonzubieta76@gmail.com

Mikel Etxeberria Garcia
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate/Mondragón, Spain.
mikel.etxeberria@ikerlan.es

Maider Zamalloa Akizu
Ikerlan Technology Research Centre,
Basque Research and Technology Alliance (BRTA)
Arrasate/Mondragón, Spain.
mzamalloa@ikerlan.es

Nestor Arana Arexolaleiba
Electronic and Computer Science
Mondragon Unibertsitatea
Arrasate/Mondragón, Spain.
Department of Materials and Production
Aalborg Universitet
Aalborg, Denmark.
narana@mondragon.edu

***Abstract*—Drones, mobile robots, and autonomous vehicles use Visual Odometry (VO) to move around complex environments. ORB-SLAM or deep learning-based approaches like DF-VO are two of the state-of-the-art technics for monocular VO. Those two technics perform correctly in outdoor scenarios but show some limitations in indoor environments. The extreme lighting conditions, non-Lambertian surfaces, or occlusion of indoor environments can disturb the visual information, and so the odometry information. Generative Adversarial Network (GAN) architectures recently proposed in the literature can help to overcome image low-light and blurring limitations. This research study aims to assess image enhancement's impact using GANS on the Visual Odometry algorithm DF-VO. Since DF-VO is also based on visual geometric information, the paper first considers the effect of two different GAN architectures in the camera's calibration. Then, the impact in the odometry information computed by DF-VO is evaluated. The preliminary results show that the reprojection error and the uncertainty of the calibration of a pin-hole-based camera do not increase significantly, and DF-VO's performance is improved.**

***Index Terms*—Image enhancement, Calibration, Visual Odometry, Deep Learning**

I. Introduction

Autonomous vehicles such as Robots, Cars and Trains use to be equipped with multiple sensors such as Inertial Measurement Unit (IMU), Laser Imaging Detection and Ranging (LiDAR), Global Positioning System (GPS), Wheel Odometry or Visual sensors. Visual sensors like RGB or RGB-Depth (RGBD) cameras in a mono or stereo settings are extensively studied since they offer rich information, which helps to understand the vehicle's surrounding environment.

On the other hand, Deep Neural Network concepts and tools are becoming more popular approaches to extract information from visual data. For example, the learning based object detection algorithm YOLO(You Only Look Once) can be used to detect traffic signs from visual information [1] in real-time. UNet is a well-known neural network that performs images semantic segmentation [2]. Visual segmentation allows us to understand the meaning of each pixel. Finally, Visual Odometry (VO) or Simultaneous Localization and Mapping (SLAM) are approaches that compute the vehicle's movement using images and SLAM, in addition to the motion, to create a map and localize itself in it [3].

All these approaches based on neural networks are practical whenever there is enough information to extract the crucial features necessary for the neural network to learn. Sometimes, these algorithms have also learning guidelines (i.e. ground truth or a loss function) from which to learn. However, there is not always enough information available.

Night driving [4], driving in varying weather conditions [5], underground railways [6], mining [7], or indoor parking [8] are some scenarios where researchers are studying how to take advantage of various types of visual sensors to compute odometry. Those scenarios are characterized by low lighting conditions like the mine tunnel or changing lighting when the train goes from the tunnel and enters the platform. We can also find specular surfaces [8], and the images are usually blurring due to poor lighting conditions.

Another issue in the scenarios mentioned above is the difficulty to create a dataset [9] with enough images to train the neural network and make it learn. Some researchers try to train the neural network using augmented data. The data quantity can be increased using traditional data augmentation techniques such as pixel colour operations, blurring using kernel filters, edge enhancement, rotation, cropping or translation. It can also be augmented using Generative Adversarial Networks (GANs) [10]. Other researchers try to enhance the images before using the pre-trained network [11] is trained for image

978-1-5386-1757-1/21/$31.00 ©2021 IEEE

recorded in good conditions, and there is no enough data to train the network with new images. In both cases, GANs seem to be a promising solution.

This research aims to study to which extend GANs can improve the monocular Visual Odometry results by enhancing low lighting and blurring images. More precisely, the effect of image enhancement using GANs on image geometrical properties will be studied.

II. Related works

A. Visual Odometry (VO)

Visual Odometry (VO) refers to using data from cameras to estimate the camera's position changes over time. We refer to monocular VO when one single camera is used to capture frames. VO is used in robotics and autonomous vehicles to calculate the robot's position displacement. Depending on the method, different monocular VO approaches can be defined. *Geometric VO* approaches rely on image geometric characteristics and perspective camera model to reconstruct ego-motion between consecutive images. One of the most standard geometric VO approaches is called ORB-SLAM2 [12] and has become the state-of-the-art comparison for all the later approaches. It is based on feature matching consecutive frames and on bundle adjustment algorithm.

Geometric monocular VO suffers from scale drift issue where scale is inconsistent and expensive global bundle adjustment algorithms are applied to refine the camera's pose. Furthermore, monocular VO algorithms have a depth-translation scale ambiguity issue.

With the improvements of Deep Learning-based Computer Vision techniques, Deep Learning in VO has started using extensively. It has led to *DL-based VO* algorithms that can solve some of these problems. Some are considered *hybrid VO* algorithms within these DL-based approaches as they mix geometric characteristics and DL-based inference.

In this context, Zhan *et al.* proposed the unsupervised VO algorithm DF-VO [17]. This algorithm is a hybrid approach that uses both geometric image information and deep learning-based depth and flow estimation to estimate the camera pose.

B. Data enhancement

Generative Adversarial Networks (GAN). GAN networks are machine learning models designed by Ian Goodfellow and his teammates in 2014 [18]. Their functioning consists of two neural networks competing with each other during their training. One network is known as the generator, and it aims to generate data capable of fooling the second network, the discriminator. At the same time, the discriminator must determine whether the generated data is real or fake. Through this competition, both networks keep improving until the generators capability to create realistic data becomes acceptable. Once the training is done, the discriminator is no longer used, and the generator is the one that works, generating new data.

This research work aims to improve images with poor or varying lighting conditions and blurring. To tackle these issues, two different GAN variations have been chosen: *EnlightenGAN* and *DeblurGAN*.

DeblurGAN. The task of image deblurring is to get the sharp version of the blurred image corrupted by some unknown blur kernel or spacially variant kernel. *DeblurGAN* [19] proposed the solution based on a conditional GAN and content loss for its learning. It achieves state-of-the-art performance in terms of structural similarity measure and visual appearance. Besides, it is 5 times faster than the closest competitor, "Deep Deblur" [20]. The DeblurGAN model used to get the sharp version of our blurred images has been a VGG19 network, pretrained on ImageNet and after that trained on corresponding blurred and sharp 256x256 patches from MS COCO dataset.

EnlightenGAN. Images captured in lousy light conditions suffer from low contrast, poor visibility and high ISO noise. Those issues challenge computer vision algorithms. The proposed solution by *EnlightenGAN* [21] is an effective unsupervised generative adversarial network. VGG-16 model pre-trained on ImageNet. The EnlightenGAN model used to improve the lightening of our images has adopted the weights of a VGG-16 model pre-trained on ImageNet. Then, because EnlightenGAN has the unique ability to be trained with unpaired low/normal light images, is has been trained with low light and normal light images from several datasets released in [22], [23] and also High-Dynamic-Ranging(HDR) sources [24], [25].

The hypothesis stated in this work proposes that the mentioned GANs may improve hybrid-VO (geometric+deep learning) approaches performance in low-lighting scenarios. The research will study first the effect of using GANs in geometric visual information using a calibration process. And then the results of the DF-VO in low-lighting trajectories with and without enhancing images using GANs.

III. Experimentation

In this section, two different experimentations will be explained; the calibration and visual odometry experimentations. In both cases, the camera used to record the datasets is a ZED Stereo Camera. The image's resolution is 1280x720 pixels at 60 Hz with an electronic synchronized rolling shutter, automatic gain and a lens aperture of F2.0. All the experimentation was done for the processing part in a workstation with an Intel i9-9900k processor with 64GB RAM and an NVIDIA RTX 2080 Ti.

The state-of-the-art monocular VO algorithms are based on minimizing the reprojection error of consecutive image frames. The reprojection error is estimated by solving the essential matrix encoding the epipolar geometry and assuming the camera satisfies the pinhole camera model. To evaluate the GAN based data enhancement architectures applicability in VO algorithms, it becomes essential to analyze the impact of GAN based data enhancement techniques in the camera parameters by following a camera calibration procedure.

Therefore, to assess the applicability of GAN-based image enhancement techniques in the VO algorithm such as DF-VO, first, its impact on the camera calibration's reprojection error will be performed.

A. Calibration experimentation

This experimentation has aimed to evaluate the impact that the used image enhancement techniques have on the reprojection error and uncertainty during the camera calibration. The calibration has been done using Matlab's calibration tool *Camera Calibrator*.

1) Calibration setup: Firstly, two primary datasets have been generated named *Day-calibration* and *Night-calibration*. The dataset *Day-calibration* consists of images captured on appropriate lighting conditions. On the other hand, the dataset *Night-calibration* is composed of the same images in the *Day-calibration* dataset but with poor light conditions. For that, the setup in figure 1 has been used, with a camera and a calibration pattern. The calibration pattern was placed in different posses inside of the visual area of the camera. In each pose, two images have been taken: One with the light of the room switched on and the second with the light turned off, simulating poor lighting conditions.

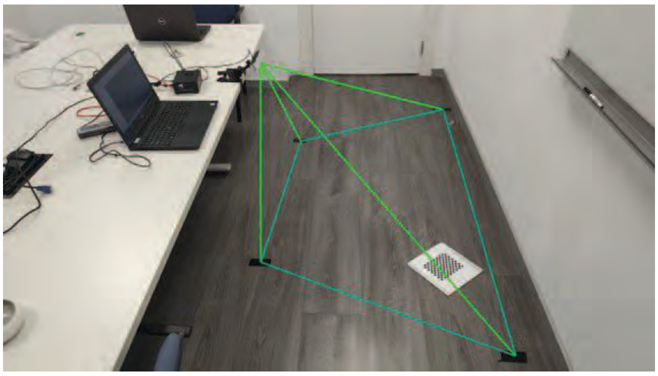


Fig. 1: Setup used to generate day-calibration and night-calibration datasets.

Once these two primary datasets have been created, two datasets have been derived from them. *DeblurGAN* has been applied to the *Day-calibration* dataset to create the *DB-calibration* dataset, and *EnlightenGAN* has been applied to the *Night-calibration* dataset to get the *Enlighten-calibration* dataset. In figure 2 can be seen the scheme of the datasets used for calibration.

2) Calibration evaluation metrics: To evaluate the calibration results, the following metrics have been used:

Mean reprojection error: Provides a qualitative measure of accuracy during calibration. A reprojection error is a distance between a pattern keypoint detected in an image used in calibration and a corresponding world point projected into the same image.

$$ReprojError_{mean} := \frac{1}{n+m}\sum_{i=1}^{n}\sum_{j=1}^{m}\sqrt[2]{(KP_{ij}\vec{p_i}-\vec{x_{ij}})^2} \tag{1}$$

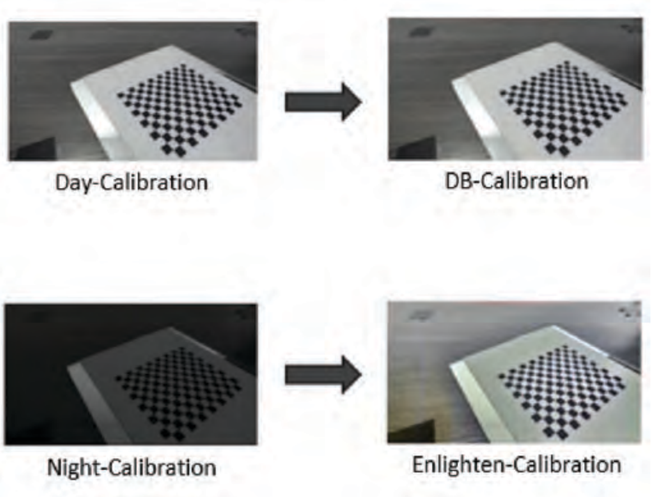


Fig. 2: Distribution of the datasets used in the dataset experimentation.

where K are the camera intrinsic parameters, P_{ij} is the camera pose, $\vec{p_i}$ are feature locations in 3D and $\vec{x_{ij}}$ its projection in a 2D image plane.

Uncertainty: It can be said that the result of a measurement is an approximation of the measured value. Therefore, to have a more realistic estimate, it is necessary to know its uncertainty.

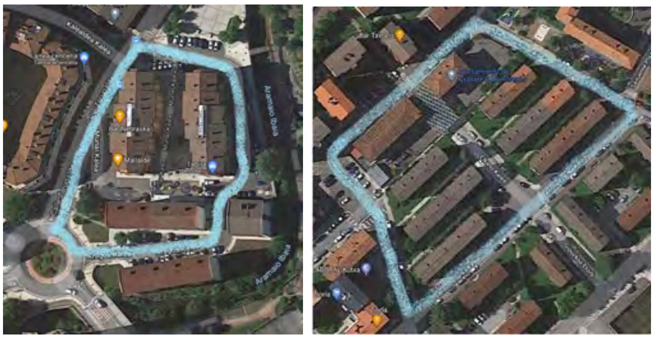
Both the reprojection error and the uncertainty have been calculated with Matlab's calibration tool.

B. Visual Odometry experimentation

1) VO setup: To evaluate the previously mentioned Data Enhancement techniques in VO algorithms, the need for a dataset with specific characteristics (i.e., consisting of dark photos or well illuminated, but with improvable sharpness) arose. Usually, VO algorithms are evaluated on standard datasets, such as KITTI [26]. After researching on most used datasets for VO evaluation [27], no standard dataset with adverse lighting conditions was identified in the autonomous vehicle sector. This led us to generate our database by making recordings of the same road trajectory during day and night. It has been chosen this road scenario because of its similarity with the KITTI dataset, which as said, is it hugely used while testing VO algorithms.

For the generation of the reference data with whom the results achieved with the VO algorithm will be compared, the state-of-the-art VO algorithm ORB-SLAM2 [12] has been used. ORB-SLAM2 is widely used as a reference in the VO community [22]–[25]. ORB-SLAM2 uses loop closure to relocalize the camera and thus improve the precision of the inferred path. For that reason, the previously mentioned recordings were done in closed paths where the starting and arrival points are the same.

Two distinct trajectories have been chosen to increase the robustness of the results and conclusions obtained from experi-



(a) Aragoa (b) Musakola

Fig. 3: Trajectories used to generate the VO experimentation dataset using a camera installed in a car.

Trajectory	Seq.	Frames	Characteristic
Aragoa	00	1150	Raw images recorded at daylight
	01	1150	Seq. 00 enhanced with *DeblurGAN*
	02	1400	Raw images recorded at night
	03	1400	Seq. 03 enhanced with *EnlightenGAN*
Musakola	04	1640	Raw images recorded daylight
	05	1640	Seq. 05 enhanced with *DeblurGAN*
	06	1480	Raw images recorded at night
	07	1480	Seq. 08 enhanced with *EnlightenGAN*

TABLE I: Full generated dataset for experimentation. The datasets contain 8 sequences, 4 from each trajectory.

mentation. The trajectories shown in figure 3 have been named after the street where they are located: Aragoa and Musakola.

Fig. 4: Camera setup for the recordings of the generated dataset.

As it can be seen in figure 4, the camera has been placed on the upper part of the front window of the car, thus preventing the front of the vehicle from invading the lower part of the recording.

Two sequences were recorded in each trajectory. One of the sequences has been recorded during the day and one at night. The sequences recorded during the day have been used to generate other sequences using the data enhancement technique mentioned above: *DeblurGAN*. The sequences recorded at night were used to create another sequences using *EnlightenGAN* data enhancement approach. Consequently, 8 sequences were generated, four for each trajectory: two original sequences and two enhanced sequences. The full gathered dataset is depicted in table I.

2) VO evaluation metrics: The evaluation of the generated enhanced datasets in DF-VO algorithm has been done using common metrics from standard datasets. Absolute Trajectory Error (ATE) is used where the root-mean-square error between estimated trajectory poses and the respective reference is calculated. For that, both trajectories are aligned through a transformation S using Horns [28] method. Given this transformation, the absolute trajectory error matrix at time i is calculated as

$$E_i := Q_i^{-1} S P_i \tag{2}$$

where P_i and Q_i are the estimated pose and ground truth pose at time i, respectively. Then the translational root-mean-square error (RMSE) over all i is calculated:

$$ATE_{trans} = \left(\frac{1}{n} \sum_{i=1}^{n} \|E_i\|^2 \right)^{\frac{1}{2}} \tag{3}$$

For VO relative evaluation, relative pose error (RPE) and average translational error (t_{err}) are also common metrics. The RPE measures the drift error per frame of the trajectory [29] . Relative pose error matrix at time step i is defined as:

$$E_i := \left(Q_i^{-1} Q_{i+1}\right)^{-1} \left(P_i^{-1} P_{i+1}\right) \tag{4}$$

where P_i and Q_i are the estimated pose and ground truth pose at time i, respectively. The total number of relative poses from a sequence with n camera poses can be calculated as $m = n - 1$. Then, the RMSE of the translational component is calculated as in the previous metric:

$$RPE_{trans} = \left(\frac{1}{m} \sum_{i=1}^{m} \|E_i\|^2 \right)^{\frac{1}{2}} \tag{5}$$

Following the evaluation criteria proposed in the KITTI evaluation benchmark, average translational error $t_{err}(\%)$ is used on sub-sequences of different lengths from the generated dataset. The error is calculated in the same way as RPE. But instead of calculating the relative error from frame to frame over all the sequence, the average error for all possible sub-sequences along $\{100, ..., 800\}$ m is used. The relative error matrix is calculated as in equation 4. Then, RMSE over all sub-sequences RPE is calculated.

IV. Results

In the following sections, the effects that image enhancement techniques(*EnlightenGAN* and *DeblurGAN*) have on the repro-jection error and uncertainty during the camera calibration, and the results of applying these data augmentation techniques on VO are shown.

1) Calibration results: The following tables II and III represent the calibration results through the previously mentioned mean reprojection error and uncertainty and the camera parameters, the focal length and the principal points.

As shown in Table I, the focal length and principal point in both *Day-calibration* and *DB-calibration* are similar. However, the *Mean reprojection error* and the *Uncertainty* increase slightly.

In Table II can be seen that the focal length and principal point in both *Night-calibration* and *Enlighten-calibration* are also similar, being the mean re-projection error and uncertainties a bit higher in *Enlighten-calibration*.

Metric	Day-calibration	DB-calibration
Frames	20	20
Mean reprojection error	0.1089	0.1208
Focal length	690.31; 689.58	690.34; 689.42
Principal point	644.96; 381.21	644.76; 380.90
Radial distortion	0.014; -0.013	0.01; -0.01
Focal length uncertainty	4.171; 4.104	4.837; 4.76
Principal point uncertainty	0.546; 2.367	0.637; 2.73
Radial distortion uncertainty	0.002; 0.002	0.002; 0.002

TABLE II: Day calibration results.

Metric	Night-calibration	Enlighten-calibration
Frames	20	20
Mean reprojection error	0.0775	0.0904
Focal length	694.058; 693.605	693.379; 692.858
Principal point	645.006; 380.768	644.871; 380.984
Radial distortion	0.015; -0.017	0.015; -0.017
Focal length uncertainty	2.558; 2.482	3.204; 3.130
Princ. point uncertainty	0.351;1.490	0.435; 1.840
Radial dist. uncertainty	0.001; 0.001	0.001; 0.002

TABLE III: Night calibration results.

The first sets of experiments show that *DeblurGAN*, and *EnlightenGAN* degrades the re-projection error and the uncertainty of the calibration parameters, that is, to the geometrical information available in the images slightly. However, geometrical information is not the only information used by the latest VO techniques. In the next section, the results of applying the image enhancement techniques(*EnlightenGAN* and *DeblurGAN*) in VO are shown.

2) VO results: As shown in table IV, in the Aragoa trajectory, on the sequences recorded at day, the minimum ATE was obtained in the original sequence 00, 18.36m, slightly lower than the sequence 01 (enhanced using *DeblurringGAN*).

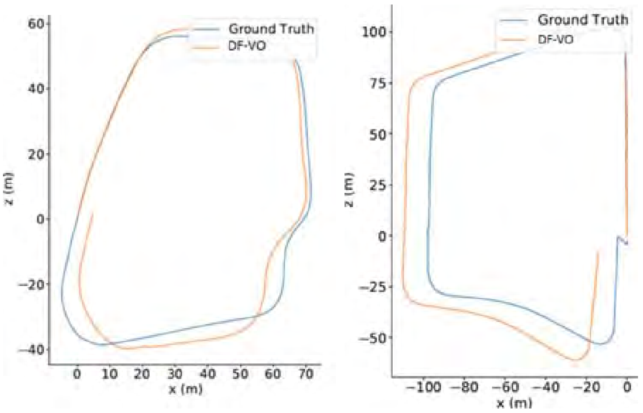
On the night sequences, ATE over a 300m run from the 02 sequence obtained was 36.21m. However, the enhanced (*EnlightenGAN*) sequence 03 has an ATE of 28.40m. Therefore, the *EnlightenGAN* enhancement algorithm has improved the results obtained in the original sequence by 7.81m.

V. Conclusions

In the Musakola trajectory, the results in terms of ATE are similar to those of Aragoa. The dataset generated by the *DeblurGAN* algorithm (sequence 05) have accumulated an ATE similar to the original 04 sequence, being this slightly lower. In the dataset generated by the *EnlightenGAN* (sequence 07), the ATE has been 12m lower than its source dataset Musakola night (sequence 06).

Trajectory	Sequence	ATE (m)	ATE (%)	RPE (m)	RPE (%)
Aragoa 300m	00	18.36	6.12	0.26	0.78
	01	19.55	6.51	0.26	0.78
	02	36.21	12.07	0.25	0.49
	03	28.40	9.46	0.24	0.49
Musakola 450m	04	15.43	3.42	0.31	0.39
	05	15.70	3.48	0.31	0.39
	06	50.78	11.73	0.35	0.46
	07	40.78	9.06	0.35	0.46

TABLE IV: Results obtained from DF-VO application in the generated dataset where Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) are measured.



(a) Sequence 00 in Aragoa. (b) Sequence 05 in Musakola.

Fig. 5: Sample comparison between DF-VO estimation and the reference created by ORB-SLAM2.

On the other hand, the relative errors (using RPE metric) were almost identical between the original dataset and its derivatives in both the Musakola and Aragoa trajectories. These results can also be seen on sample trajectories in figure 5, where the reference created using state-of-the-art ORB-SLAM2 and the result obtained with DF-VO are compared.

In this paper, GAN based image enhancement architectures in an unsupervised monocular visual odometry algorithm called DF-VO is evaluated. The evaluated GAN techniques *DeblurGAN* and *EnlightenGAN*, aim to enhance the image's blurring and lighting conditions. The images captured by a camera installed in an autonomous vehicle (car, train, tram), being a moving element, may include low-light and blurring that can disturb the visual information.

State-of-the-art monocular VO algorithms are based on minimizing the reprojection error of consecutive frames captured by the camera. The error is estimated by solving the essential matrix, which depends on the intrinsic camera parameters, and assuming the camera satisfies the pinhole camera model. In this study, the enhanced images calibration procedure is pursued to assess the three GAN architecture's effect in the camera's calibration. The experimental results show that the analyzed GAN architectures do not disturb the camera calibration parameters significantly. The reprojection error and the uncertainty of the camera's intrinsic parameters (optical centre and focal length) being slightly worse are still aligned with the optimal calibration parameters provided by the manufacturer. Therefore, the experimental calibration results support GAN-based enhancement architectures in unsupervised monocular VO algorithms.

The selected visual odometry algorithm for GAN based enhancement architectures evaluation is DF-VO. DF-VO is an unsupervised hybrid visual odometry algorithm based on deep learning and geometric properties. A proprietary database containing the characteristics required by the GAN architectures (low-light, blurring and improvable sharpness) was created by a monocular camera installed in a car. Two close-loop car routes were defined, and the same routes with daylight and nightlight were recorded.

The experimental results show *EnlightenGAN* architectures improve the DF-VO performance in low-light scenarios. A possible explanation is that *EnlightenGAN* could increase the number of characteristic points used by DF-VO even if image uncertainty is slightly worse in the enhanced images. The ATE value decreases in Aragoa route from 12.07 to 9.46, and it reduces from 11.73 to 9.06 in the route Musakola. The performance obtained by *DeblurGAN's* architecture is more limited. The experimental results observed a similar performance when evaluating the DF-VO algorithm in daylight enhanced images. Note these results support the conclusion obtained in the images' calibration procedure enhanced by the GAN architectures.

ACKNOWLEDGMENT

This work was supported in part by the Basque Government through BIKAINTEK2018 program and CAF Signalling. The authors would also like to acknowledge support by European Union's H2020-WIDESPREAD project no. 857061 "Networking for Research and Development of Human Interactive and Sensitive Robotics Taking Advantage of Additive Manufacturing - R2P2". Finally, the authors will thanks to Daniel Maestro Watson for his input on uncertainties.

REFERENCES

[1] Choodowicz, Ewelina, Pawe Lisiecki, and Piotr Lech. "Hybrid algorithm for the detection and recognition of railway signs." International Conference on Computer Recognition Systems. Springer, Cham, 2019.

[2] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[3] Poddar, Shashi, Rahul Kottath, and Vinod Karar. "Evolution of visual odometry techniques." arXiv preprint arXiv:1804.11142 (2018).

[4] H. Zhang, I. Ernst, S. Zuev, A. Börner, M. Knoche and R. Klette, "Visual Odometry and 3D Point Clouds Under Low-Light Conditions," 2018 International Conference on Image and Vision Computing New Zealand (IVCNZ), Auckland, New Zealand, 2018, pp. 1-6, doi: 10.1109/IVCNZ.2018.8634769

[5] Wang, Ke and Chen, Junlan and Ren, Fan. (2020). Approaches, Challenges, and Applications for Deep Visual Odometry: Toward to Complicated and Emerging Areas. IEEE Transactions on Cognitive and Developmental Systems. PP. 10.1109/TCDS.2020.3038898.

[6] M. Etxeberria-Garcia, M. Labayen, M. Zamalloa and N. Arana-Arexolaleiba, "Application of Computer Vision and Deep Learning in the railway domain for autonomous train stop operation," 2020 IEEE/SICE International Symposium on System Integration (SII), Honolulu, HI, USA, 2020, pp. 943-948, doi: 10.1109/SII46433.2020.9026246.

[7] C. Kanellakis and G. Nikolakopoulos, "Evaluation of visual localization systems in underground mining," 2016 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 2016, pp. 539-544, doi: 10.1109/MED.2016.7535853.

[8] Senbo Wang, Jiguang Yue, Yanchao Dong, Shibo He, Haotian Wang, Shaochun Ning, "A synthetic dataset for Visual SLAM evaluation", Robotics and Autonomous Systems 124 (2020) 103336

[9] Wang, Ke and Chen, Junlan and Ren, Fan. (2020). Approaches, Challenges, and Applications for Deep Visual Odometry: Toward to Complicated and Emerging Areas. IEEE Transactions on Cognitive and Developmental Systems. PP. 10.1109/TCDS.2020.3038898.

[10] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).

[11] Eunah Jung and Nan Yang and Daniel Cremers, "Multi-Frame GAN: Image Enhancement for Stereo Visual Odometry in Low Light", 3rd Conference on Robot Learning, Osaka, Japan (CoRL 2019).

[12] Mur-Artal, Raul, and Juan D. Tardós. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.

[13] Mur-Artal, Raúl, and Juan D. Tardós. "Visual-inertial monocular SLAM with map reuse." IEEE Robotics and Automation Letters 2.2 (2017): 796-803.

[14] Taketomi, Takafumi, Hideaki Uchiyama, and Sei Ikeda. "Visual SLAM algorithms: a survey from 2010 to 2016." IPSJ Transactions on Computer Vision and Applications 9.1 (2017): 1-11.

[15] Delmerico, Jeffrey, and Davide Scaramuzza. "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots." 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018.

[16] Yang, Nan, et al. "D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[17] Zhan, Huangying, et al. "DF-VO: What Should Be Learnt for Visual Odometry?." arXiv preprint arXiv:2103.00933 (2021).

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative Adversarial Networks, ". Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.

[19] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin , J. Matas "DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks, ". 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.

[20] S. Nah, T. Hyun, K. Kyoung, and M. Lee "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring". 2016.

[21] Y. Jiang, X. Gong, D. Liu, Y. Cheng, C. Fang, X. Shen, J. Yang, P. Zhou, and Z. Wang "EnlightenGAN: Deep Light Enhancement without Paired Supervision". 2021 IEEE.

[22] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: a raw images dataset for digital image forensics. In Proceedings of the 6th ACM Multimedia Systems Conference, pages 219–224. ACM, 2015.

[23] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. arXiv preprint arXiv:1808.04560, 2018.

[24] Nima Khademi Kalantari and Ravi Ramamoorthi. Deep high dynamic range imaging of dynamic scenes. ACM Trans. Graph, 36(4):144, 2017

[25] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. IEEE Transactions on Image Processing, 27(4):2049–2062, 2018

[26] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012.

[27] Etxeberria-Garcia, Mikel, et al. "Application of Computer Vision and Deep Learning in the railway domain for autonomous train stop operation." 2020 IEEE/SICE International Symposium on System Integration (SII). IEEE, 2020.

[28] Horn, Berthold KP. "Closed-form solution of absolute orientation using unit quaternions." Josa a 4.4 (1987): 629-642.

[29] Sturm, J., Burgard, W., and Cremers, D., "Evaluating egomotion and structure-from-motion approaches using the tum rgb-d benchmark," in Proc. of the Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RJS International Conference on Intelligent Robot Systems (IROS), (2012).

A.5 Visual Odometry in challenging environments: an urban underground railway scenario case

The following paper has been accepted for publication in IEEE Access.

Etxeberria-Garcia, M., Zamalloa, M., Arana-Arexolaleiba, N., and Labayen, M. (2022). Visual Odometry in challenging environments: an urban underground railway scenario case. *IEEE Access*.

Digital Object Identifier

Visual Odometry in challenging environments: an urban underground railway scenario case

MIKEL ETXEBERRIA-GARCIA[1], MAIDER ZAMALLOA[1], NESTOR ARANA-AREXOLALEIBA[2,3], and MIKEL LABAYEN[4,5]

[1]Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA). P.º J.M. Arizmendiarrieta, 2. 20500 Arrasate/Mondragón. (e-mail: {mikel.etxeberria,mzamalloa}@ikerlan.es)
[2]MGEP, Mondragon Unibertsitatea, Loramendi Kalea, 4 20500 Arrasate-Mondragon Gipuzkoa, Spain (e-mail: narana@mondragon.edu)
[3]Department of Materials and Production, Aalborg University | Fibigerstræde 16, Room 4-209 | 9220 Aalborg East | Denmark
[4]CAF Signalling, Juan F. Gilisagasti Kalea, 4, 20018 Donostia, Gipuzkoa
[5]Faculty of Informatics, UPV/EHU, Manuel Lardizabal Ibilbidea, 1, 20018 Donostia, Gipuzkoa, Spain (e-mail: mikel.labayen@ehu.eus)

Corresponding author: Mikel Etxeberria-Garcia (e-mail: mikel.etxeberria@ikerlan.es).

This work was supported in part by the Basque Government through BIKAINTEK2018 program and CAF Signalling. In collaboration between Ikerlan, CAF Signalling and Mondragon Unibertsitatea. We would also like to thank Euskotren and Eusko Trenbide Sarea.

ABSTRACT Localization is one of the most critical tasks for an autonomous vehicle, as position information is required to understand its surroundings and move accordingly. Visual Odometry (VO) has shown promising results in the last years. However, VO algorithms are usually evaluated in outdoor street scenarios and do not consider underground railway scenarios, with low lighting conditions in tunnels and significant lighting changes between tunnels and railway platforms. Besides, there is a lack of GPS, and it is not easy to access such infrastructures. This research proposes a method to create a ground truth of images and poses in underground railway scenarios. Second, the EnlightenGAN algorithm is proposed to face challenging lighting conditions, which can be coupled with any state-of-the-art VO techniques. Finally, the obtained ground truth and the EnlightenGAN have been tested in a real scenario. Two different VO approaches have been used: ORB-SLAM2 and DF-VO. The results show that the EnlightenGAN enhancement improves the performance of both approaches.

INDEX TERMS Visual Odometry, Autonomous vehicles, Computer vision, Data enhancement, Simultaneous localization and mapping, Image processing, Railway domain

I. INTRODUCTION

VISUAL ODOMETRY (VO) is a particular case of odometry based on Computer Vision (CV), where the position and motion information are acquired through camera images [1]. VO algorithms aiming to derive localization data through visual sensors are usually evaluated and compared by reference standard datasets such as KITTI [2, 3] and EuRoC-MAV [4]. This situation leads solutions adapted to the visual characteristics contained on those scenarios with adequate lighting conditions (good illumination and similar lighting conditions in subsequent frames), relatively sufficient textures and Lambertian surfaces. However, few algorithms, datasets, and benchmarks can be found in challenging scenarios with varying light conditions, low illumination, low textures, or non-Lambertian surfaces.

For instance, one of the latest benchmark challenges in visually challenging odometry is the Subterranean Challenge (SubT), organized by the Defense Advanced Research Projects Agency (DARPA). Perceptually challenging scenarios and tasks were stated in this challenge, such as navigation through tunnel systems, cave networks, or urban underground environments. The participating teams presented several approaches [5, 6, 7, 8] to study the robotics autonomy in underground scenarios exploration and navigation. These works emphasize the complexity of localization and navigation in underground environments due to their perceptually-degraded conditions. They also emphasize on the importance of field testing.

The railway domain is also moving towards the *Intelligent Transportation Systems* (ITS) and the *Advanced Driving Assistance Systems* (ADAS) industry. A train that implements autonomous operations requires accurate localization estimation to carry out operations as precise stop operation or coupling successfully. Algorithms applied in urban underground railway scenarios must deal with significant light changes from tunnel areas to platforms, with insufficient illumination and low textures in tunnels.

In this context, the application of state of the VO algorithms and data enhancement techniques was analyzed in a perceptually challenging driving car scenario [9]. The results showed that the Generative Adversarial Network (GAN)-based image enhancement methods can improve the performance achieved by state-of-the-art VO solutions.

In this paper, an analysis of state-of-art VO algorithms is performed and the use of a data enhancement method in underground railway VO solutions is evaluated. Algorithms applied in these scenarios must deal with significant light changes from tunnel areas to platforms, with insufficient illumination and low textures in tunnels. Therefore, an image enlightening technique is integrated to improve the results of state-of-the-art VO algorithms.

A dataset with challenging characteristics is really needed in order to evaluate VO performance in such scenarios. From an analysis of datasets used in CV for localization (datasets labeled with 6-DoF pose), no standard dataset of the railway domain was found; hence, an ad-hoc underground railway dataset generation was pursued.

The following section (II) includes a literature review of the main VO algorithms, a description of the applied enlightening data enhancement technique, and a list of reference VO datasets. Section III depicts the urban underground railway dataset generation process. Then, the results of state-of-art VO algorithms in the underground railway dataset and the influence of an enlightening technique are shown in sections IV and V, respectively. Finally, some conclusions are drawn in section VI.

II. LITERATURE REVIEW

A. VISUAL ODOMETRY

The term Visual Odometry was first introduced by Niester *et al.* [10] proposing a technique to estimate camera motion using RANSAC [11] outlier refinement method and tracking extracted features across the frames. Previously, feature matching was done just in consecutive frames. Later works have shown that VO methods might perform as well as wheel odometry while the cost of cameras is much lower compared to wheel sensors [1].

The VO research community started from the robotics domain to, later, focus on the localization in other subdomains. In this context, different types of vehicles from distinct sub-domains and diverse characteristics have been studied, such as, cars [12][13], trains [14], or lately UAVs [15].

Depending on the algorithm used to estimate odometry data, VO techniques can be classified as learning-based and geometry-based [16, 17]. *Geometry-based VO* is usually divided into appearance-based VO (also referred to as direct), feature-based VO, and a hybrid approach that mixes the two of them.

Direct VO techniques operate directly on intensity values. In feature-based VO methods features are extracted from the image and a tracking-matching process is done. Feature-based methods have good accuracy, are robust in dynamic scenes, and can deal with variances in viewpoint [18]; however, in contrast to direct methods, feature-based techniques are inadequate in low texture areas. However, the performance of direct VO algorithms degrades if the dataset is not photometrically calibrated and is sensitive to geometric distortions as those induced by the camera speed [19]. Furthermore, as mentioned in [20], direct methods require a constant irradiation appearance between matched pixels, which hinders its application in some scenarios.

Geometry-based VO approaches rely on image geometric characteristics and camera model to reconstruct the ego-motion between consecutive frames. One of the most standard geometric VO approach is ORB-SLAM2 [21]. It is based on the ORB [22] feature matching and a bundle adjustment algorithm. It is the reference geometric solution in the VO community [23, 24, 25, 26, 19, 27, 28].

Geometry-based VO is reliable and accurate under favorable conditions, when there are enough illumination and textures to make the feature matching among consecutive frames. As stated in [29], monocular VO experiences a scale drift issue and global bundle adjustment algorithms needs to be applied. Furthermore, monocular VO algorithms have a depth-translation scale ambiguity issue [30].

Stereo geometry-based VO works have been also targeted lately. Semi-direct visual odometry (SVO) [31] is one of the most predominant approaches among direct monocular and stereo VO algorithms. It uses a probabilistic mapping method to estimate ego-motion and explicitly models outlier measurements. In 2017, Wang *et al.* presented Stereo Direct Sparse Odometry (Stereo DSO) [19], a method for VO estimation from stereo cameras based on the previously proposed monocular DSO algorithm [32]. Lately, Koestler *et al.* presented TANDEM [33], a SLAM system that estimates ego-motion based on a direct VO pipeline and deep multi-view stereo.

The expansion of Deep Learning-based Computer Vision techniques carried the emergence of *Deep Learning-based VO* solutions. Learning-based VO/vSLAM algorithms usually rely on learning parts of a standard VO/vSLAM pipeline or designing end-to-end trainable algorithms for ego-motion estimation.

One of the first and most relevant learning-based VO algorithms was PoseNet proposed by [34] Kendall *et al.*, a robust and real-time monocular re-localization system based on an end-to-end trained CNN. This approach was later improved by introducing loss functions based on geometry and scene reprojection error [35]. Following this end-to-end pose estimation networks, DeepVO [36] was published, a solution that infers camera poses directly in an end-to-end manner from a sequence of RGB frames through a supervised Deep Recurrent Convolutional Neural Network (RCNN).

Some research works have tried to adapt traditional non-learning approaches into Deep Learning pipelines. Brachmann *et al.* introduced DSAC (Differentiable Sample Consensus) [37] algorithm based on previously proposed RANSAC [11]. They applied DSAC in a camera localization solution, learning an end-to-end camera localization pipeline.

However, most of the research works from the literature emphasize the importance of an accurate depth and flow estimation for VO/vSLAM. Depth information is crucial for the localization as it enables the inference of the scene geometry from 2D images. Moreover, it allows scale recovery [38] and the distinction of foreground and background points, allowing a better environment understanding. Together with depth estimation, the optical flow estimation is also a critical component of some VO/vSLAM algorithms as it models the motion between consecutive images. Therefore, most of learning-based VO/vSLAM algorithms have focused on learning depth and flow estimation for the pose inference process.

Following this research line, several works have focused the depth estimation [39],[40],[41]. In 2018, Zhan *et al.* presented Depth-VO-Feat [42], where stereo training was introduced to reduce the spatial and temporal photometric error. At the same time, DVSO was presented by Yang *et al.* [29], introducing deep depth predictions in Direct Sparse Odometry (DSO). D3VO [43] algorithm was also proposed in this direction, including the uncertainty estimation with camera pose and depth.

Zhan *et al.* proposed the unsupervised VO algorithm DF-VO [17]. This algorithm applies a deep learning-based depth and flow estimation, and, geometric image information to estimate the camera pose. As shown in [17], DF-VO outperforms most learning-based state-of-the-art algorithms in standard datasets.

Some works have proposed loss functions to handle challenging scenario characteristics. Yin et al. proposed GeoNet [44], to increase robustness towards outliers and non-Lambertian surfaces. After GeoNet, more works were proposed in this direction [45],[46].

However, as mentioned in [47], literature VO solutions have limitations in challenging scenarios that contain insufficient illumination and textures, or, variable lighting conditions. Literature VO solutions, as they are adapted to the characteristics of standard datasets, require sufficient illumination and enough textured surfaces for a correct feature matching. A good illumination allows motion extraction from images, as pixel displacement can not be accurately estimated otherwise. Therefore, the lighting issue needs to be handled in scenarios that contain low illumination or varying illumination conditions. These are the conditions that face the urban underground railway scenario.

DF-VO and *ORB-SLAM2* have been selected from the literature review as reference VO algorithms. As stated before, the DF-VO algorithm outperforms most learning-based state-of-the-art algorithms, while ORB-SLAM2 is the most referenced geometric algorithm. Moreover, these algorithms represent two distinct types of VO algorithms (learning-based and geometric). Both solutions can use mono-vision or stereo-vision camera frames as input. The stereo-vision input was chosen for the analysis, as stereo-vision solutions keep the real-world scale, i.e. the predictions are directly aligned to a real-world scale.

B. DATA ENHANCEMENT FOR VISUAL ODOMETRY IN CHALLENGING ENVIRONMENTS

In order to afford the scenario limitations of VO in challenging environments, the application of a data enhancement technique was considered. In this work, the data enhancement process is dedicated to the lighting limitations of the target domain. It aims to reduce the impact of the drastic lighting conditions found in the underground railway scenario.

In this paper, the work published in [9] is extended. In the previous work the application of *EnlightenGAN* [48] data enhancement approach in an outdoor driving car scenario with varying lighting conditions was evaluated. This previous research was focused on a driving car scenario where the lighting conditions of the underground railway domain where replicated driving by night. The results showed that the performance of DF-VO algorithm is improved when EnlightenGAN is applied in the recorded frames.

EnlightenGAN is based on machine learning models proposed by Ian Goodfellow *et al.* [49]. The algorithm uses an unsupervised Generative Adversarial Network (GAN) pretrained on the ImageNet dataset [50] and then trained on several datasets [51, 52, 53, 54] to improve input image lighting.

EnlightenGAN was previously used for several tasks such as image reconstruction [55], photo exposure correction [56], image quality assessment [57] or illumination enhancement [58]. However, to our knowledge, the use of data enhancement methods to handle specific problems of VO methods in such challenging scenarios has not been researched yet.

In this paper, the application of EnlightenGAN in the underground railway domain when using geometric and hybrid VO solutions is evaluated. The study aims to explore if EnlightenGAN technique can afford the lighting limitations of reference VO approaches (DF-VO and ORB-SLAM2). The evaluation procedure and results are detailed in section V.

C. DATASETS FOR UNDERGROUND RAILWAY VISUAL ODOMETRY

In this work, a propietary dataset is generated as no standard or reference railway dataset fitted to the underground railway scenario was identified. Table 1 resumes the reference datasets used by starte-of-the art VO approaches.

Most state-of-the-art VO approaches are evaluated in the standard KITTI [2, 3] vision benchmark [36, 82, 42, 29, 17, 43]. This benchmark includes several datasets for tasks like VO, optical flow estimation, 3D object detection, or 3D tracking. The data is captured from a moving car in outdoor urban scenarios, and they provide datasets and evaluation

Dataset	Domain	Sensor configuration	Pose ground truth	Environment
Cambridge Landmarks [34]	Handheld sensor	Monocular	SfM	outdoors
7-scenes [59]	Handheld sensor	RGB-D	MoCap	indoors
BigSFM [60]	Handheld sensor	Monocular	GPS	outdoors
ICL-NUIM [61]	Handheld sensor	RGB-D	SLAM	indoors
ADVIO [62]	Handheld sensor	Stereo/IMU	IMU	in/outdoors
OIVIO [63]	Handheld sensor	Stereo/IMU	Total station	in/outdoors
Rawseeds [64]	Robot	Stereo/IMU	GPS	in/outdoors
SUN3D [65]	Robot	RGB-D	SfM	indoors
TUM-VI [66]	Robot	Stereo/IMU	MoCap	in/outdoors
TUM-RGB-D SLAM [67]	Robot	RGB-D	MoCap	indoors
TUM-Monocular VO [68]	Robot	Monocular	LSD-SLAM/MoCap	in/outdoors
NavVis [69]	Robot	Monocular	GPS	indoors
MIT Stata [70]	Robot	Stereo/RGB-D/Laser	Laser	indoors
The Wean Hall [71]	Robot	Stereo/IMU/Laser/Wheel odometry	GPS	in/outdoors
RGB-D SLAM [72]	Robot	RGB-D	MoCap	indoors
ETH3D [73]	Robot	Stereo/RGB-D/Laser/IMU	MoCap/SfM/LIDAR	in/outdoors
NCLT [74]	Segway	Stereo/IMU/Laser	GPS/IMU/Laser	in/outdoors
KITTI [2, 3]	Car	Stereo/IMU/Laser	GPS/IMU	outdoors
Málaga Urban [75]	Car	Stereo/IMU/Laser	GPS	outdoors
Oxford RobotCar [76]	Car	Stereo/Laser	GPS	outdoors
Ford Campus [77]	Car	Stereo/Laser/IMU	GPS	outdoors
KAIST Urban [78]	Car	Stereo/IMU	GPS/Laser	outdoors
Nordland [79]	**Railway**	**Monocular**	**GPS**	**outdoors**
Zurich Urban [80]	MAV	Monocular/IMU	GPS	outdoors
EuroC/MAV [4]	MAV	Stereo/IMU	MoCap/Laser	indoors
MVSEC [81]	Multi Vehicle	Stereo/IMU/Laser	GPS/MoCap/Laser	in/outdoors

* MoCap=Motion Capture System. SfM=Structure From Motion

TABLE 1: Referenced datasets for Computer Vision-based VO approaches application and evaluation ordered by domain or motion type.

metrics for each task. However, as the KITTI odometry dataset contains images from an outdoor environment with good lighting conditions, it is not adequate to evaluate the VO algorithms in the pursued scenario. Among the other analyzed datasets, it should be noted that only one database (Norland [79]) covers the railway domain; however it only covers outdoor scenarios, which is also out of the scope of this research work. Searching for a publicly available VO dataset from an indoor urban railway domain, no dataset was found. Following the idea that the evaluation of the VO approaches that have previously been evaluated in standard datasets is essential to adapt the algorithms to other industrial scenarios. Therefore, the generation of a proprietary database was considered.

The data for a proprietary dataset can be collected from different sources: from real scenarios or simulated environments. Real environment datasets are based on real-world scenarios, and therefore, the performance of algorithms can be effectively evaluated in the target scenario. However, the database generation in real-world scenarios increases recording and processing time, effort, and cost. In addition, it also depends on the access and permission to make the recordings in the target scenario.

Simulated environments can overcome these problems. The drawback of simulated environments is that it can not be assured that an algorithm trained and validated in a simulated environment will perform the same way in a real-world scenario. As stated in [83], all the challenging conditions inherent to underground environments can not be recreated in virtual scenarios.

Consequently, and as a real-world underground railway scenario was accessible, a proprietary dataset was generated from a real underground railway scenario. The definition, generation and validation processes of the proprietary *CAF* dataset is explained in the next section III.

III. URBAN UNDERGROUND RAILWAY DATASET GENERATION

The proprietary (*CAF*) was generated for the evaluation of VO algorithms in underground railway scenarios. The sensor set validation and camera calibration procedure was done by generating a complementary dataset (*CarDriving*) in an urban driving car domain. *CarDriving* dataset generation is described in [9].

The *CAF* dataset was recorded in an underground scenario in the railway *Line 3* of Euskotren-Bilbao. The line is composed by seven stations from Matiko to Kukullaga and it has a whole track length of 5.8km. It contains poor lighting conditions in tunnel areas and significant light changes in platform areas. Furthermore, the images captured in the tunnels contain repetitive and light dependent textures, and therefore, they are challenging for feature extraction algorithms. Figure 1 shows two frames of this scenario: (a) tunnel frame and (b) platform frame.

The camera was placed in the front of the train, inside the driving cabin according to the safety requirements of the railway domain. Figure 2 shows the camera placement in the active cabin.

The recording camera is a ZED Stereo Camera. The image's resolution is 1280x720 pixels at 30 Hz with an

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3187209

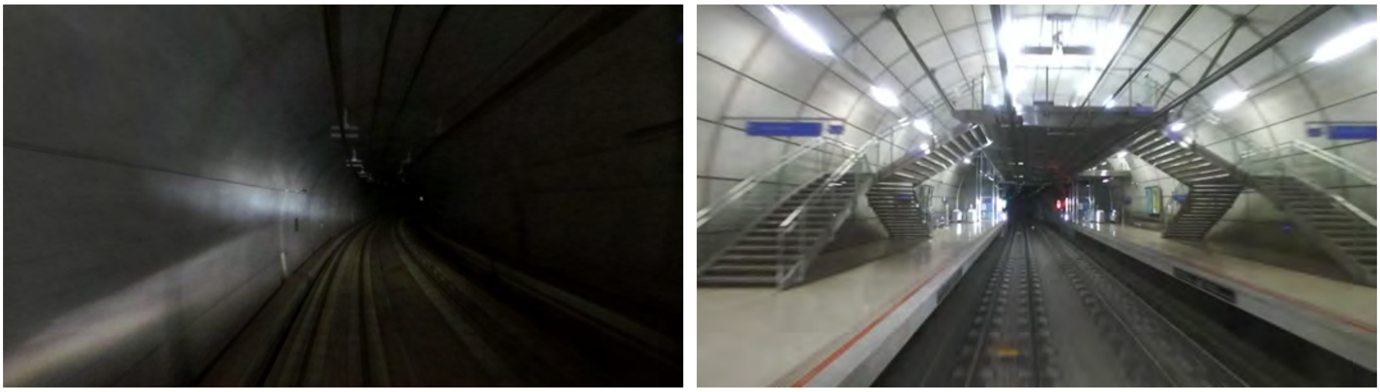


FIGURE 1: The *CAF* dataset's tunnel and platform areas where the poor light conditions and textureless areas can be appreciated.

electronic synchronized rolling shutter, automatic gain and a lens aperture of F2.0.

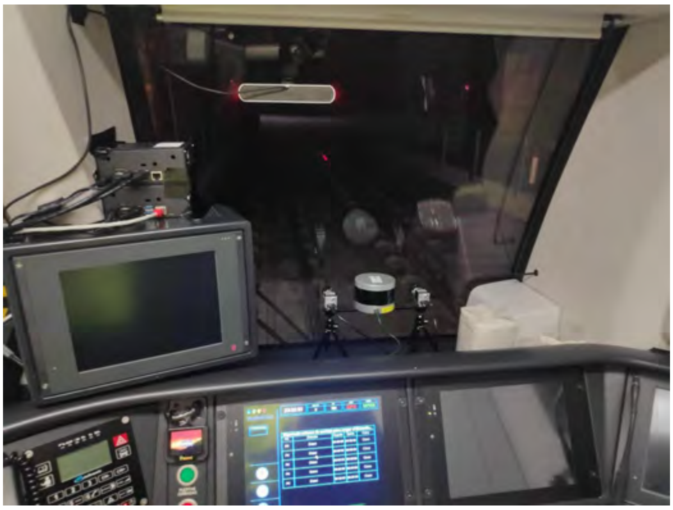


FIGURE 2: Camera setup for *CAF* dataset, placed in the cabin of a train moving through an underground urban railway scenario.

A. CAF DATASET

The dataset is composed by 19 sequences captured in the two directions of the rail Matiko-Kukullaga. A sequence is a record that begins at one station and ends in the stations the train stops. A 6-DoF pose is estimated for each captured frame. The dataset format follows the standard KITTI odometry dataset format and naming convention. The frames are rectified RGB color images stored with lossless compression using 8-bit PNG files.

The camera calibration parameters and the poses are stored in files specified by the KITTI format [3]. Each row of the pose file contains the first three rows of a 4x4 homogeneous pose matrix flattened into one line. The homogeneous pose matrix p_n can be represented as:

$$p_n = [r_n|tr_n] = \begin{bmatrix} r11 & r12 & r13 & x_n \\ r21 & r22 & r23 & y_n \\ r31 & r32 & r33 & z_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where r_n and tr_n are the rotation matrix and the translation matrix of the n-th frame, respectively. The translation component of the pose matrix follows the right-hand rule when defining axes in a 3D space (x-axis forward, y-axis right and z-axis up).

The dataset generated in this domain is represented in table 2 where the recorded sequences, recording direction, the arrival station for each sequence, the number of frames, and the track length of each sequence are depicted. The entire set of sequences yields 65.384 frames, with varying speed and length.

Direction	Arrival station	Sequence	Frames	Length (m)
Matiko	Otxakoaga	01_50	3048	1420
	Txurdinaga	01_53	1977	699
	Zurbaranbarri	01_54	2663	1029
		03_49	6700	3148
	Zazpikaleak	02_22	3260	1011
	Uribarri	01_15	2724	903
		02_25	2639	
		03_54	5904	1913
	Matiko	01_17	2532	505
		02_27	2505	
Kukullaga	Uribarri	01_31	2140	524
	Zazpikaleak	01_33	2830	979
	Zurbaranbarri	01_35	2494	1007
		03_36	6560	2449
	Txurdinaga	01_37	2550	1032
	Otxarkoaga	01_39	2126	695
		03_36	4493	1729
	Kukullaga	01_40	4095	1405
		03_44	4144	
TOTAL			65384	23261

TABLE 2: *CAF* dataset resume with recorded sequences, the direction of the sequences, arriving station for each sequence, frame quantity, and sequence length.

B. GROUND TRUTH GENERATION ALGORITHM DATA SOURCES

In general, the ground truth of VO datasets is generated using a GPS sensor [3, 75, 76, 77, 78] (refer to Table 1). But, the GPS signal is unavailable in underground zones like the urban underground railway domain. Thus, a method that computes the 6-DoF pose of each frame from the train ERTMS/ETCS ATP data, geodetic map coordinates, and railway infrastructure gradient profile data was defined and implemented (see figure 3).

The algorithm first estimates (x,y) positions based on geodetic coordinates, then z is added through the gradient profile. Afterwards the (x,y,z) translation data is estimated for each frame by using ERTMS ATP data, and, finally, the rotation data of each pose is calculated.

1) Geodetic coordinates

The geodetic coordinates are represented by a pair (ϕ, λ) expressing *Latitude (Lat.)* and *Longitude (Lon.)* in decimal degrees. These coordinates use an ellipsoid to approximate the the earth's surface locations [84].

In this research, the geodetic coordinates define the coordinates followed by the trains in the target railway and have been extracted from a Geomap called ÖPNVKarte [85]. This Geomap contains public data that includes worldwide public transport facilities on a uniform map with information concerning several transport methods such as train, railway, ferry or bus. It is derived from OpenStreetMap [86], an initiative to create and provide accessible geographic data (i.e. street maps, etc.). It also contains railway-related information, such as platforms, stop positions, and routes.

The entire trajectory of an underground train in L3 extracted from ÖPNVKarte is shown in figure 4. As stated before, the trajectory of L3 is made up of seven stations in the route Kukullaga - Matiko, where some route positions, the station entrances, and train stop positions of each station are known in geodetic coordinates. However, the frequency of the camera is higher than the geodetic coordinates defined in the Geomap, and, therefore, a method based on ERTMS ATP data has been designed and implemented in order to generate the poses of the frames that were recorded between the geodetic coordinates.

The geodetic coordinates must be transformed from 3D plane to a 2D plane to assign an equal-area (x,y) position to each geodetic coordinate. Figure 5 shows a trajectory sample in geodetic coordinates and the generated equal-area (x,y) coordinates. In the ground truth generation algorithm, an equal-area (x,y) coordinate refers to tr_x and tr_y components of a 6-DoF pose.

2) Railway gradient profile

The railway gradient profile provided by the railway infrastructure managers, defines how the slope of the railway varies in predefined sections and allows the estimation of the height (z) for each 6-DoF pose. For that, a height profile can be constructed with this gradient profile. The initial height is initialized as 0, and then the height for each 1m section is calculated using the Equation 1.

$$h(d_n) = h(d_{n-1}) + (0.01 * grad_n) \tag{1}$$

where h refers to height, d_n refers to 1m railway sections and $grad_n$ is the gradient value corresponding to that section from the gradient profile. Figure 6 shows the obtained railway gradient profile of the whole L3 railway.

3) ATP data: train's dynamics and speed data

The ERTMS/ETCS ATP train speed estimation process is based on redundant wheel encoder and radar sensor in order to get a safe and accurate estimation. By using these sensors, the ATP subsystem embedded in the train estimates the train position in the track, i.e. the distance traveled from an station or a beacon of the track. Track beacon position or inter-beacon distance is predefined and known by railway infrastructure managers, even by the ATP subsystem, and therefore the ATP train position is re-adjusted when a beacon signal is received obtaining a precise estimation. The 6-DoF pose estimation of each frame is made by synchronizing the ATP system monitoring process with the image recording process as both are installed in the train. The objective of this process is to obtain a synchronized train position information for each frame. The data monitored from the ATP system is the following one:

- *timestamp* (s): time measured in the Coordinated Universal Time (UTC) standard read from the train's internal clock.
- *linear position estimation* (cm): distance traveled by the train from a previous station.
- *train speed* (m/s): train speed calculated by ATP.
- *train acceleration* (cm/s^2): train acceleration calculated by ATP.
- *train stopped*: boolean reflecting whether the train has reached stopping point or not.

All those variables are extracted from a ATP monitoring proprietary application that monitors ATP data with a frequency of 128000 Hz. The data acquisition frequency higher than the camera frequency (30Hz), and, consequently, they have been synchronized and a pose estimated for each frame.

C. ESTIMATE POSES OF AN INTERVAL THROUGH A BACKWARD DATA SYNCHRONIZATION BASED ON TIMESTAMP

The main idea of the synchronization algorithm is the estimation of poses in the trajectory sections between the known (x,y) positions obtained by transforming the known geodetic coordinates. These known (x,y) positions define the trajectory, but they are not enough for camera frequency and, therefore, more poses must be estimated between them. The *interval* has been defined to represent the idea of the trajectory sections, and it is a straight line between two consecutive known (x,y) positions. The estimated poses are located in the intervals. Figure 7 represents the intervals,

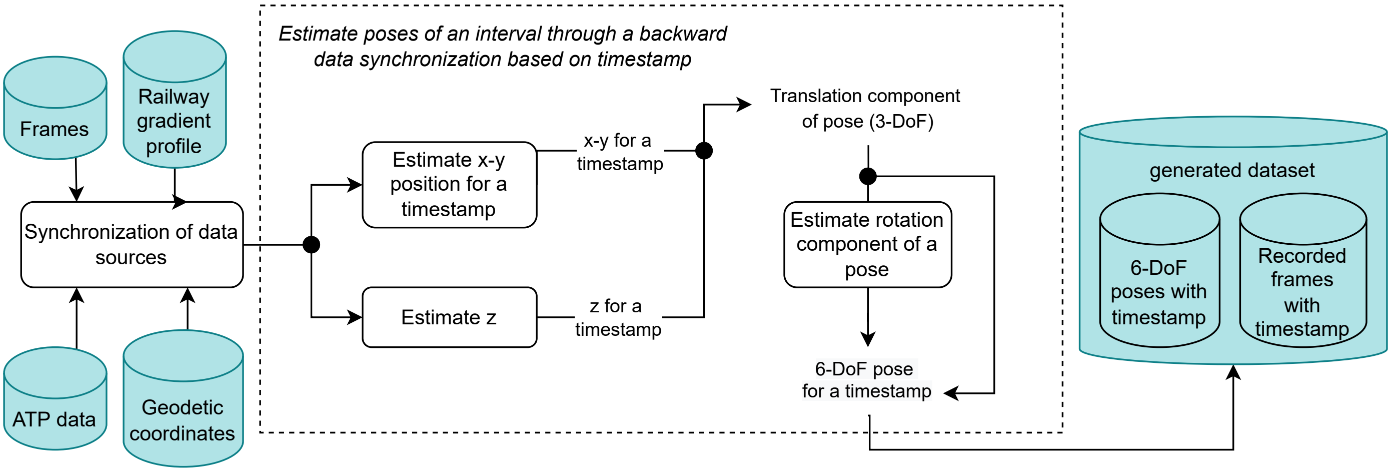

Frames
Railway gradient profile
Synchronization of data sources
ATP data
Geodetic coordinates
Estimate poses of an interval through a backward data synchronization based on timestamp
Estimate x-y position for a timestamp
x-y for a timestamp
Estimate z
z for a timestamp
Translation component of pose (3-DoF)
Estimate rotation component of a pose
6-DoF pose for a timestamp
generated dataset
6-DoF poses with timestamp
Recorded frames with timestamp

FIGURE 3: Diagram of the algorithm processes, with the data sources and the outputs.

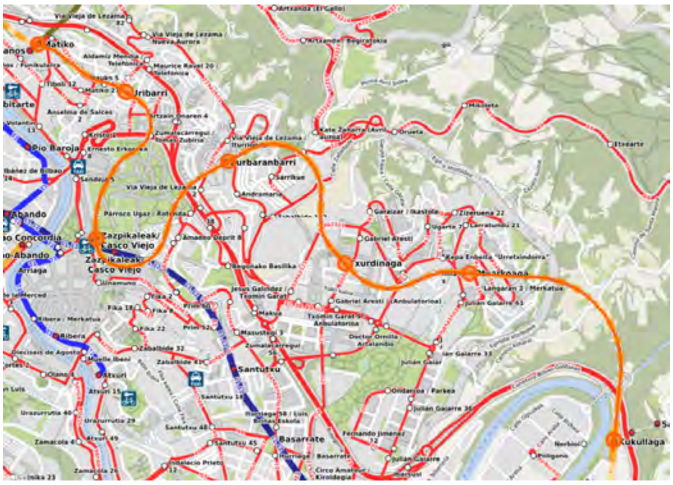


FIGURE 4: Line 3 railway extracted from ÖPNVKarte map [85]. Each circle represents one station from Line 3.

known (x,y) positions and estimated poses in the railway. The main concepts of the ground truth generation algorithm are described in 1.

As the data sources are synchronized at the sequence ending, from now on, the ground truth generation is done in a backward data synchronization process of an *interval* based on the images timestamps. The last (x,y) position, last image and ATP data are taken for a given interval and the poses for all timestamps in that interval are estimated. Then, the poses of the following interval are estimated by taking the last (x,y) position and the last image of the previous interval as the initial position.

However, the train speed is variable and, therefore, the distribution of these poses can not be linear in different intervals. The total number of poses within the whole sequence should match the record frame amount.

Algorithm 1 Ground truth data generation algorithm

```
Input: Given an interval ($i$) defined as a straight line between two known (x,y) positions
Phase 1 - Synchronize last (x,y) position, last image and ATP data of an interval
if $i = 0$ then                                                  ▷ First interval
    Last image ← $SSIM > threshold$                              ▷ SSIM [87]
    Last $(x_i, y_i)$ position ← given in the interval definition
    ATP data ← $train\_stopped = 1$
else                                                             ▷ Following intervals
    Last image, last $(x_i, y_i)$ position and ATP data ← taken from $i - 1$
end if
Phase 2 - Estimate poses on an interval through a backward data synchronization based on timestamps
Input: $V_n$: train speed, $a_n$: train acceleration, $t$: timestamp, $h$: height profile, $d_n$: linear position estimation
Estimate translation component of poses ($tr_n$)
    a: $(x_n, y_n) \leftarrow f(v_n, a_n, t)$                    ▷ Eqn. 2
    b: $z_n \leftarrow h(d_n)$                                   ▷ Eqn. 1
Estimate rotation component of poses ($r_n$)
    a: $r_n \leftarrow g(tr_{n-1}, tr_n)$                        ▷ Eqn. 3, 4, 5
```

1) Synchronize last (x,y) position, last image and ATP data

The first step is to synchronize the different data sources using the last (x,y) position, last image and ATP data. The algorithm generates ground-truth poses for each recorded sequence using the position where the train has stopped as origin. For that, first the image where train stops (last image of the sequence) must be estimated. When there is motion, the similarity between consecutive frames is very low, however the similarity increases when the train has stopped. Due to the similarity of the frames corresponding to the train stopping point, the last frame is selected using the Structural Similarity Index (SSIM) [87]. SSIM is one of the most

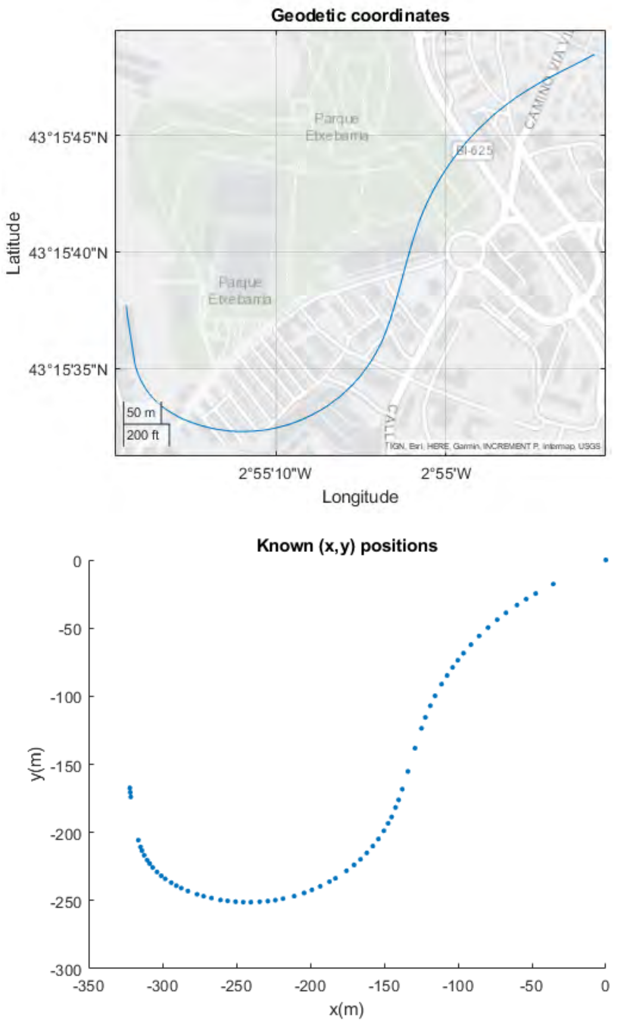


FIGURE 5: Transformation of a given sequence from railway defined by geodetic coordinates into equal-area (x positions.

standard algorithms for image quality assessment [57], and therefore, for image similarity measure. It has shown that can outperform other common image similarity measurements as MSE [88] and has been previously referenced [89]. The SSIM measures the luminance, contrast, and structure of two given images and returns a similarity value between them.

Also, it only requires a starting optimization phase where the threshold is selected. Furthermore, the index was used to find just the first image within the threshold in each sequence, which gives a little number of results totally. Although SSIM is sensitive to image distortions, the environment being static, and the view fixed enables the SSIM application in underground railway scenarios.

The threshold was selected by exploratory testing. A predefined threshold was stated and iterated it until a SSIM threshold that best fitted to the lighting conditions of the scenario was identified. In this case a $SSIM > 0.965$ has been used as similarity threshold at the train stopping point.

The last (x,y) coordinates refer to the train stopping position; therefore, this coordinate pair and the last image are already synchronized. Finally, ATP monitored data is synchronized using the *train stopped* variable.

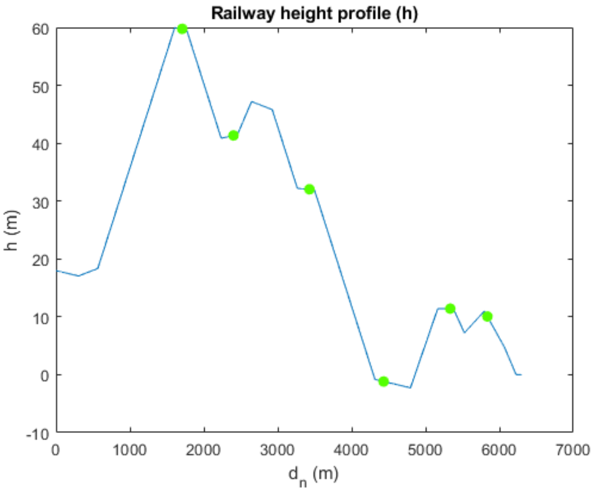


FIGURE 6: Results of height generation process. Height profile (h) is generated from gradient profile provided by

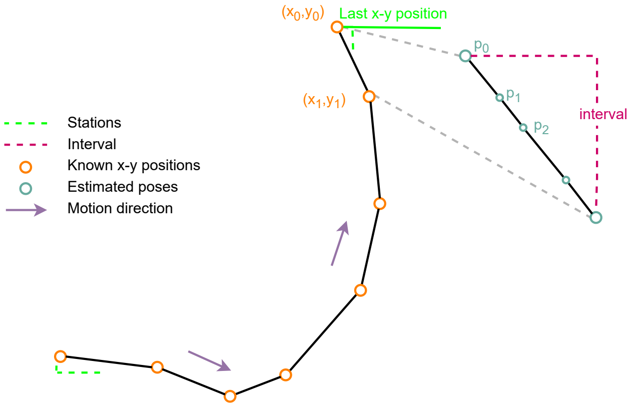


FIGURE 7: Railway with the known (x,y) positions, the intervals and estimated poses.

2) Estimate poses of an interval through a backward data synchronization based on timestamps

A ground truth pose is generated for each recorded image in an interval using a backward synchronization process based on the timestamp. This process has two steps; first, the translation component is estimated, and then, the rotation is calculated from that translation.

Estimation of translation component. Translation component $T = \{tr_0, tr_1, ..., tr_m\}$ is defined as a set containing all the 3-DoF poses ($tr_n = [x_n, y_n, z_n]$) of an interval where n is the pose number ($0 \leq n \leq m$) and m is the total number of poses for that interval. For the translation component of a pose, first, the (x,y) position is estimated, and then the height (z) is added. The translation is estimated by taking an initial (x,y) position and calculating the motion to the next one using the ATP data *train speed* and *train acceleration*. The translation between two consecutive (x,y) positions in a straight line that forms the interval can be calculated using *Uniformly Accelerated Motion (UAM)* equations. This estimation is possible because it is considered that the poses follow a motion in a straight line and with a constant acceleration between them. Equation 2 shows the application of UAM equations in this case.

$$d_n = v_{n-1}t + \frac{1}{2}a_{n-1}t^2 \tag{2}$$

where t refers to the timestamp, v_n and a_n refer to ATP data train speed and acceleration respectively. The initial (x,y) translation component is set as $[0, 0]$.

After calculating the (x,y) positions, the z or height is estimated using the height profile estimated from the gradient profile and ATP data. The railway height profile can be synchronized with the train stopping point, and therefore, with the first (x,y) position.

Then, previously calculated (x,y) positions can be used to extract the Euclidean distance traveled from position to position. Each pose's height (z) is calculated using traveled distances and the height profile. Therefore, after height estimation, the translation component of a pose has been estimated with respect to a timestamp.

Estimation of rotation component. Rotation component $R = \{r_0, r_1, ..., r_m\}$ is defined as a set containing all the rotation matrices (r_n) within an interval where n is the pose number ($0 \leq n \leq m$) and m is the total number of poses for that interval calculated in the previous steps.

To calculate the rotation component r_n for each translation tr_n the transformation between two consecutive orientation vectors or_{n-1} and or_n is estimated. or_n defines the orientation of the train in tr_n and represents the vector between consecutive translations tr_{n-1} and tr_n. It is calculated as shown in 3:

$$or_n(tr_{n-1}, tr_n) = (x_n - x_{n-1}, y_n - y_{n-1}, z_n - x_{z-1}) \tag{3}$$

where x, y and z represent the translation components of tr_{n-1} and tr_n. Then, using the axis-angle representation, the transformation between consecutive orientation vectors or_{n-1} and or_n can be calculated. For that, first the orientation vectors are normalized by dividing their value with the Euclidean norm (vector magnitude) $\|or_n\|$ of each vector (Eqn. 4) to align them at the same origin. The Euclidean norm can also be defined as the Euclidean distance of a vector from the origin to a point.

$$normalize(or_n) = \frac{or_n}{\|or_n\|} \tag{4}$$

Then, the Euclidean norm of the cross product between the normalized consecutive orientations is estimated to get the axis. Finally, the rotation component is estimated using the inverse tangent function as shown in equation 5, where the angle between the orientations vectors is calculated trough the dot product:

$$r_n = acos(\frac{\|or_n \times or_{n-1}\|}{or_n \cdot or_{n-1}}) \tag{5}$$

where $acos$ refers to the inverse tangent function and or_{n-1} and or_n to two consecutive orientation vectors. This rotation estimation method accumulates an error relative to the previous estimations. However, as the train is tied to the rails, the trains' orientation is always fixed, and the orientation estimation is not critical.

The previously calculated translation component is added to the newly calculated rotation component to obtain the target 6-DoF ground truth pose. This is done by following the representation in equation III-A.

Once all the poses from a given interval have been estimated, the next interval is taken and the process is repeated until all the intervals of a sequence have been covered.

IV. VO APPLICATION IN URBAN UNDERGROUND RAILWAY ENVIRONMENT

In this section the application of DF-VO and ORB-SLAM2 in the CAF dataset is evaluated.

In the following subsection, the standard VO evaluation metrics are explained. Then, the experimentation setup is described. Finally, the experimental results are discussed.

A. VO EVALUATION METRICS

The metrics used to evaluate the performance of the experiments are the following: Absolute Trajectory Error – *ATE* [72], Relative Pose Error – *RPE* [72], Average Translational Error – t_{err} and Average Rotational Error – r_{err}.

All the sequences were transformed with a 6-DoF Umeyama alignment [90], a standard alignment method used in most VO and SLAM evaluation benchmarks. [2]. A 6-DoF alignment is recommended to evaluate shape similarities of trajectories [91].

Given this transformation, ATE evaluates the global consistency of an estimated trajectory compared to the ground-truth trajectory. The RPE measures the drift error for each pose of the trajectory and the rotation and the translation components are calculated separately.

Finally, following KITTI evaluation benchmark criteria, the Average Translational Error (t_{err}) and the Average Rotational Error (r_{err}) are calculated on sub-sequences of different lengths. These errors measure the average relative pose error at a fixed distance. The sub-sequences length in meters is (100,200,...,800) because the error for smaller sub-sequences was large and hence biased the evaluation results.

B. EXPERIMENTATION SETUP

These experiments extend the evaluation done at [9], where ORB-SLAM2 and DF-VO were evaluated in an outdoor urban car driving scenario. In those experiments, the bad lighting conditions were replicated by car driving recordings in the night.

DF-VO implementation [92] flow-weights and depth estimation deep models were selected from the authors' trained models. The flow model is trained by the authors in the synthetic dataset Scene Flow [93].

To handle the non-deterministic nature of the ORB-SLAM2 algorithm, each sequence is run five times, and the median accuracy is evaluated as proposed by authors in [21]. The VO evaluation is done using the *KITTI Odometry Evaluation Toolbox* [17].

C. VO RESULTS IN CAF DATASET

Table 3 shows the results of DF-VO and ORB-SLAM2 in the CAF dataset. Figures 8 and 9 represent the results depicted in table 3. The visual representation can be found in Figure 9.

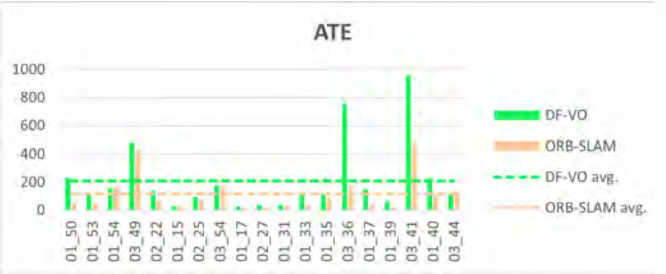


FIGURE 8: ATE of DF-VO and ORB-SLAM2 application on the generated *CAF* dataset.

Previously, DF-VO and ORB-SLAM2 were evaluated in the KITTI Odometry dataset; however, KITTI does not contain those perception challenges as it contains considerably different properties related to the sequence length and visual characteristics. Results in *CAF* dataset show that the errors of both algorithms are higher than those found in the KITTI dataset. The RPE for DF-VO is 0.038 and 0.339 in KITTI dataset and *CAF* dataset, respectively. While for ORB-SLAM2, RPE measures are 0.130 and 0.353.

In the case of the ATE, the error of DF-VO in KITTI dataset is 6.344 while in the *CAF* dataset is 210.517. For ORB-SLAM2, the ATE is 26.48 and 115.754 in KITTI dataset and *CAF* dataset, respectively.

It can be seen that ORB-SLAM2 outperforms DF-VO in this challenging scenario, where the sequences are longer than the standard KITTI dataset. If the *CAF* sequences are shortened to just platform areas where the lighting challenges are more limited, and more similar to the lighting conditions of the KITTI dataset, the errors are reduced to similar values (see Table 4) of executing DF-VO, and ORB-SLAM2 in KITTI dataset [21, 17]. For instance, DF-VO achieves an RPE (m) of 0.027 in KITTI dataset and 0.049 in shortened *CAF* dataset. ORB-SLAM2 achieves an ATE of 9.464 in KITTI dataset while 4.113 is achieved in shortened *CAF* dataset. Furthermore, the same behavior as in KITTI dataset is observed: DF-VO performance is higher than ORB-SLAM2. These results seem to support that the challenging scene conditions hinder the application of VO algorithms in such scenarios.

Results are visually shown in figure 10. In the case of DF-VO, a scale misalignment can be appreciated as the shape of most estimated trajectories is similar to the ground truth shape, but a dimensionality error appears.

As mentioned in [17], geometry-based VO algorithms as ORB-SLAM2 suffer from a scale drift when ideal visual conditions are not met. In the case of DF-VO, being a hybrid algorithm, the scale may be wrongly estimated due to issues related to the geometric characteristics of the underground visual domain or deep-learning training process. The estimation error of the learning part of the algorithm could be reduced by training the deep models in the target scenario.

Nevertheless, these results require an adaptation of reference VO solutions to increase the performance in the underground railway domain. Image enhancement techniques or solutions based on the fusion of different odometry sensors could provide the precision required by autonomous train operations.

V. ENLIGHTENGAN IN VO APPLICATION

This section explores the application of the image enhancement technique EnlightenGAN in ORB-SLAM2 and DF-VO algorithms.

VO algorithms are based on minimizing the reprojection error of consecutive frames captured by the camera. The error is estimated by solving the essential matrix, which depends on the intrinsic camera parameters, and assuming the camera satisfies the pinhole camera model. In a previous work, the enhanced images calibration procedure was pursued to assess the EnglithenGAN architecture's effect on the camera's calibration. The experimental results showed that the GAN architecture did not significantly disturb the camera calibration parameters. Therefore, it was concluded that VO algorithms could be applied directly to the dataset enhanced by EnlightenGAN.

In the following section the enhanced dataset generation, the experimental configuration, and, finally, the results are explained.

A. ENHANCED DATASET GENERATION: ENLIGHTENCAF

The CAF dataset enhanced by EnlightenGAN is named *EnlightenCAF*. Figure 11 shows the result of the enhancement in the same tunnel zone frame as in figure 1.

The same algorithm configuration from *CAF* dataset experimentation has been used. The enhancing inference model is composed of pretrained weights from original authors.

B. RESULTS IN ENLIGHTENCAF

In the previous work [9], the experimental results showed that *EnlightenGAN* improves the DF-VO performance in low-light car scenarios. In this case, the same behavior was

Algorithm	Record	14_11_2021 (->Matiko)									
	Seq	01_50	01_53	01_54	03_49	02_22	01_15	02_25	03_54	01_17	02_27
DF-VO	t_{err} (%)	80	52.97	85.74	86.27	77.94	64.09	94.52	111.55	56.61	55.69
	r_{err} (°/100m)	13.21	21.25	29.92	18.48	35.88	24.43	33.5	41.71	21.34	21.33
	ATE	230.66	106.38	157.46	478.76	135.36	29.11	94.27	175.64	26.1	36.39
	RPE (m)	0.402	0.232	0.354	0.423	0.269	0.236	0.314	0.34	0.132	0.133
	RPE (°)	0.156	0.135	0.156	0.176	0.124	0.13	0.157	0.143	0.057	0.062
ORB-SLAM2	t_{err} (%)	68.79	54.93	177.16	125.85	80.28	55.03	94.26	136.06	51.89	53.88
	r_{err} (°/100m)	5.95	19.19	50.42	17.2	25.6	20.17	31.97	39.34	15.21	14.24
	ATE	56.58	44.98	169.39	435.36	72.35	26.71	74.61	177.23	15.61	17.1
	RPE (m)	0.34	0.192	0.641	0.646	0.277	0.204	0.302	0.371	0.116	0.122
	RPE (°)	0.081	0.092	0.429	0.264	0.125	0.104	0.11	0.121	0.065	0.064

Algorithm	Record	14_11_2021 (->Kukullga)									Avg. Err.
	Seq	01_31	01_33	01_35	03_36	01_37	01_39	03_41	01_40	03_44	
DF-VO	t_{err} (%)	63.64	135.76	175.28	148.11	136.82	58.1	110	93.32	99.36	**93.9879**
	r_{err} (°/100m)	17.81	28.84	29.64	23.51	27.19	21.01	22.21	31.19	41.2	26.50789
	ATE	38.38	104.98	119.08	754.45	150.64	62.88	957.69	226.86	114.75	210.5179
	RPE (m)	0.183	0.467	0.583	0.541	0.594	0.192	0.365	0.332	0.35	0.35389
	RPE (°)	0.088	0.13	0.132	0.116	0.166	0.103	0.111	0.103	0.147	0.125895
ORB-SLAM2	t_{err} (%)	58.49	145.36	185.8	101.98	155.7	51.05	125.01	94.92	96.31	100.6711
	r_{err} (°/100m)	16.71	22.53	24.31	20.11	17.51	17.15	18.82	10.41	10.41	**20.9079**
	ATE	30.67	38.47	88.96	172.66	36.31	22.28	478.87	103.74	137.45	**115.754**
	RPE (m)	0.167	0.498	0.649	0.388	0.672	0.154	0.362	0.311	0.312	**339053**
	RPE (°)	0.09	0.099	0.113	0.11	0.113	0.079	0.097	0.07	0.07	**0.12084**

TABLE 3: DF-VO and ORB-SLAM2 application evaluation using standard VO evaluation metrics: Average Translational Error (t_{err}), Average Rotational Error (r_{err}), ATE and RPE. The sequences are organized by the direction they are recorded. The average errors for all 19 sequences are calculated, and the best result is in bold.

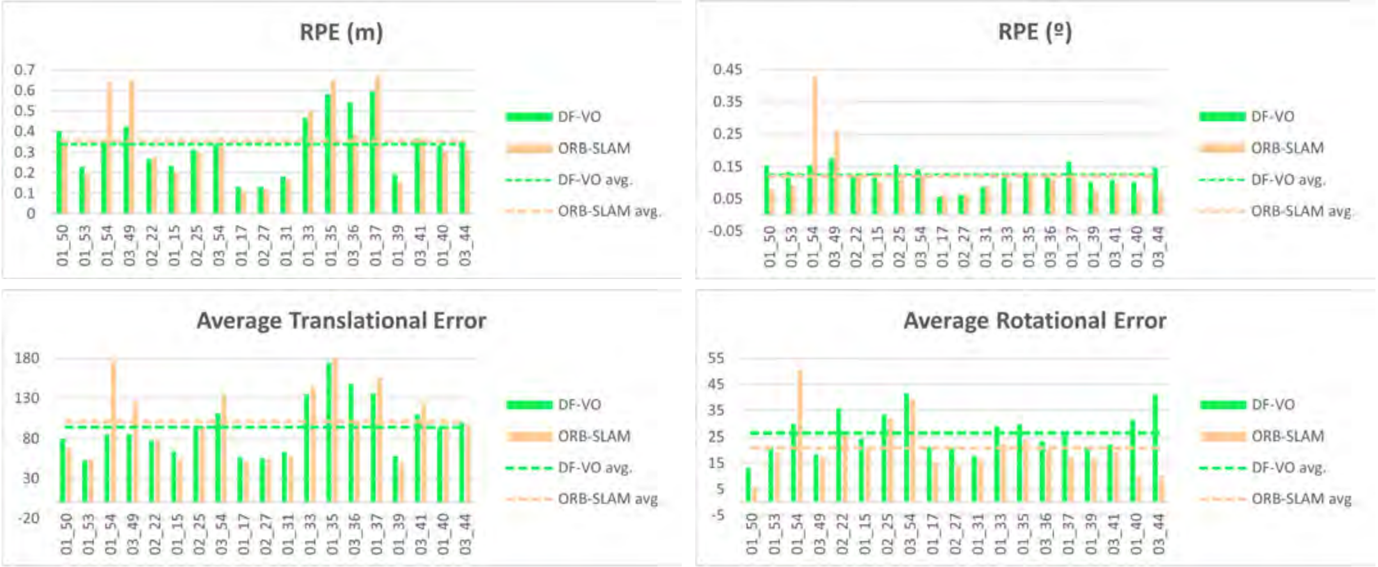


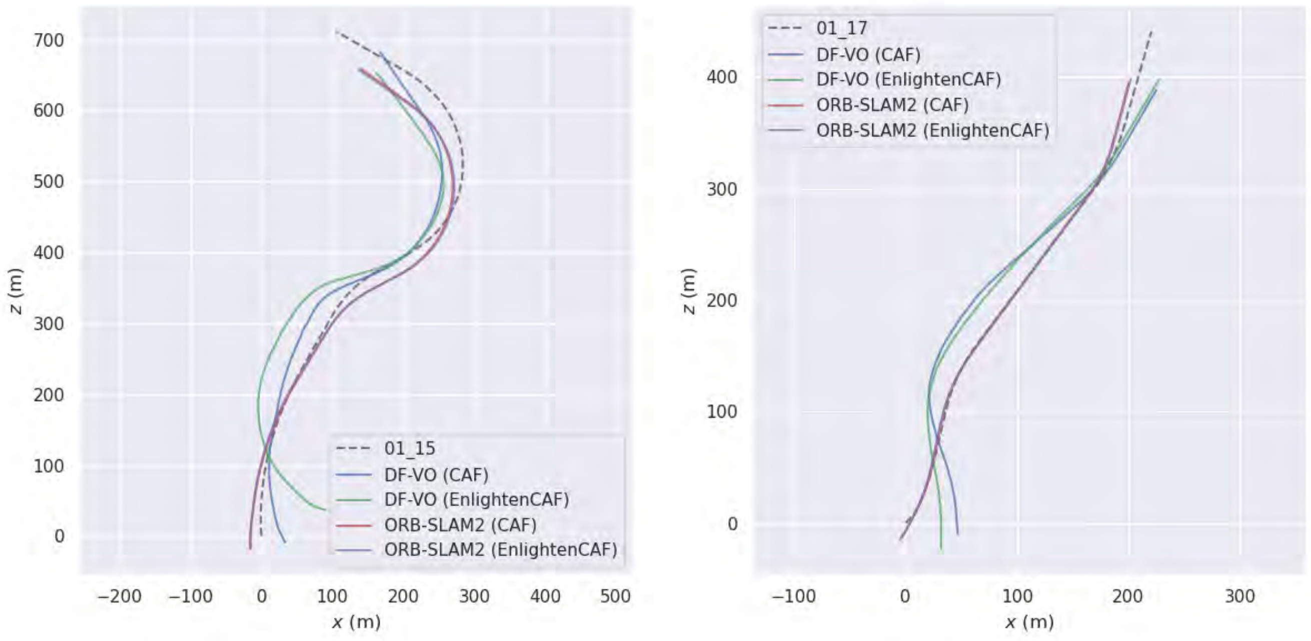
FIGURE 9: Comparison of relative VO evaluation metrics when applying DF-VO and ORB-SLAM2 algorithms in CAF datasets. Translational and rotational components of relative errors are shown separately.

confirmed: quantitative results show that EnlightenGAN reduces the VO errors for both algorithms. Figure 12 shows the reduction in the mean ATE and mean RPE of both algorithms for all the sequences in *EnlightenCAF*.

A relative ATE reduction of 24.89% and 20.20% is observed, respectively, when DF-VO and ORB-SLAM2 are applied in the enhanced sequences. Figure 13 shows RPE, t_{err} and r_{err} evaluation metrics in EnlightenCAF dataset.

In the case of RPE, DF-VO algorithm obtains a relative improvement of 1.97% and 4.74% for translation and rotation components, respectively. ORB-SLAM2 gets a relative improvement of 14.59% for the RPE translation component and a relative improvement of 18.55% for the rotation component. t_{err} and r_{err} present a relative reduction of 0.22% and 4.16% when applying DF-VO, and a relative reduction of 3.63% and 9.31% when applying ORB-SLAM2.

Figure 10 shows a result comparison of DF-VO and ORB-SLAM2 in the sequences of CAF and EnlightenGAF. As in *CAF* dataset, it can be seen that the algorithms can estimate the shape of the *EnlightenCAF* trajectories. However, a scale



(a) Sequence 01_15

(b) Sequence 01_17

FIGURE 10: Comparison of ORB-SLAM2 and DF-VO application on two sample sequences in both CAF and EnlightenCAF datasets and the ground truth for each trajectory.

Algorithm	Metric	Avg. Err
DF-VO	t_{err} (%)	18.520
	r_{err} (°/100m)	6.975
	ATE	2.298
	RPE (m)	0.049
	RPE (°)	0.037
ORB-SLAM2	t_{err} (%)	19.484
	r_{err} (°/100m)	14.681
	ATE	4.113
	RPE (m)	0.0798
	RPE (°)	0.126

TABLE 4: Average standard VO errors in *CAF* dataset when reducing the sequences to platform areas without lighting constraints.

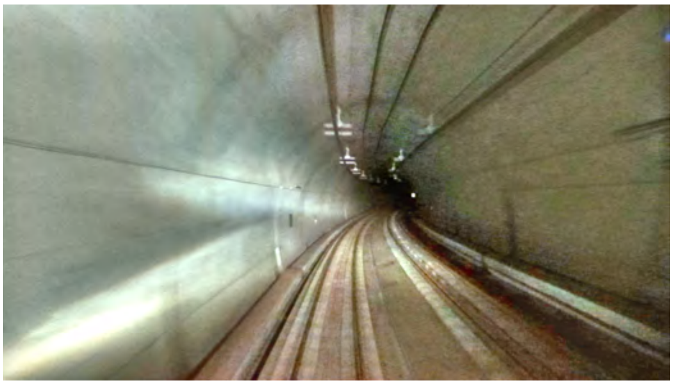


FIGURE 11: A frame from the *CAF* dataset enhanced by EnlightenGAN.

underestimation problem appears again. Furthermore, DF-VO results show that the rotation estimation is affected in the *EnlightenCAF* dataset.

The results demonstrate that EnlightenGAN improves VO algorithms performance in the underground railway domain. Furthermore, the relative error is reduced more for the geometric-based VO algorithm, while absolute error is reduced more in the learning-based algorithm.

However, as in the *CAF* dataset, the errors continue being higher than the results obtained by the algorithms in the KITTI dataset. Therefore, an affection of lighting conditions of the scenario can still be appreciated. This affection could be related to scale underestimation problems found in both algorithms, especially in the hybrid DF-VO.

Additionally, when evaluating the VO algorithms, it has been seen that the dispersion of the poses estimated by ORB-SLAM2 in different runs is reduced when enhancing the frames with EnlightenGAN.

The dispersion of poses among different executions of ORB-SLAM2 has been evaluated using standard metrics [94]. These metrics include the *variance* (σ^2) and the *Coefficient of Variation* (cv).

The evaluation procedure has been to run ORB-SLAM2 five times in each dataset, the original *CAF* and the enhanced *EnlightenCAF*. Figure 14 shows the results of applying ORB-SLAM2 five times for a given sequence (*01_54*) in the *CAF* and the enhanced *EnlightenCAF* datasets. It can be seen that the distribution of the poses through the trajectory is more constant in the enlightened dataset.

From the results, it can be seen that enlightening the

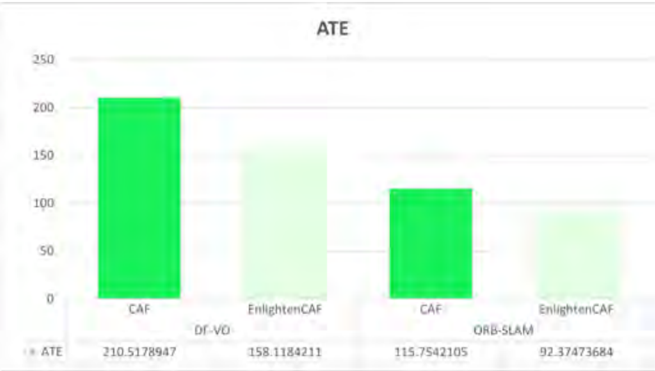


FIGURE 12: Comparative of ATE when applying DF-VO and ORB-SLAM2 algorithms in CAF and EnlightenCAF datasets.

datasets with *EnlightenGAN* increases the VO performance and tends to reduce ORB-SLAM2 dispersion. An analysis of the trouble spots in the dispersion results could better understand the high dispersion in such frames and detect further possible improvements for VO algorithms in such scenarios.

VI. CONCLUSION

This paper has presented a method to create a ground truth database for underground railway scenarios, where the GPS is unavailable, or the access to the infrastructure is not easily granted. The ground truth data generation is based on camera frames, ERTMS/ETCS ATP data, the railway gradient profile map, and geodetic coordinates of the target railway. Second, it has proposed to enhance image lighting conditions with EnlightenGAN, which can be used with any state-of-the-art VO. Finally, it has presented the result of the experiment performed within a real urban underground railway scenario. The scenario was characterized by varying lighting conditions (tunnel vs. platform), low illumination (in tunnels), or texture-less areas that challenged the state-of-the-art VO algorithms. The experiments were performed using two VO approaches: geometric (ORB-SLAM) and hybrid (DF-VO). The results show that the data enhancement increases the performance of both VO algorithms, reducing the translational error by at least 18%.

Future research proposes to apply the proposed dataset generation method and image enhancement algorithm in more underground railway scenarios. Sensor fusion is also a promising research direction. It is expected that the inclusion of new sensors will reduce uncertainty and increase accuracy, which will be welcome for autonomous train operations requiring higher localization accuracy (e.g., precise train stop operation).

REFERENCES

[1] Khalid Yousif, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics". In: *Intelligent Industrial Systems* 1.4 (2015), pp. 289–311. ISSN: 2363-6912. DOI: 10.1007/s40903-015-0032-7.

[2] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 3354–3361.

[3] Andreas Geiger et al. "Vision meets robotics: The kitti dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[4] Michael Burri et al. "The EuRoC micro aerial vehicle datasets". In: *The International Journal of Robotics Research* 35.10 (2016), pp. 1157–1163.

[5] Christopher G Atkeson et al. "Achieving reliable humanoid robot operations in the DARPA robotics challenge: team WPI-CMU's approach". In: *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*. Springer, 2018, pp. 271–307.

[6] Tomáš Rouček et al. "Darpa subterranean challenge: Multi-robotic exploration of underground environments". In: *International Conference on Modelling and Simulation for Autonomous Systesm*. Springer, Cham. 2019, pp. 274–290.

[7] Kamak Ebadi et al. "LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 80–86.

[8] Ali Agha et al. "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge". In: *arXiv preprint arXiv:2103.11470* (2021).

[9] Jon Zubieta Ansorregi et al. "Image Enhancement using GANs for Monocular Visual Odometry". In: *2021 IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM)*. IEEE. 2021, pp. 1–6.

[10] D. Nistér, O. Naroditsky, and J. Bergen. "Visual Odometry". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004*. 2004, p. 1.

[11] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[12] Gideon P Stein, Ofer Mano, and Amnon Shashua. "A robust method for computing vehicle ego-motion". In: *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511)*. IEEE. 2000, pp. 362–368.

[13] Koichiro Yamaguchi, Takeo Kato, and Yoshiki Ninomiya. "Vehicle ego-motion estimation and moving object detection using a monocular camera". In: *18th International Conference on Pattern Recognition (ICPR'06)*. Vol. 4. IEEE. 2006, pp. 610–613.

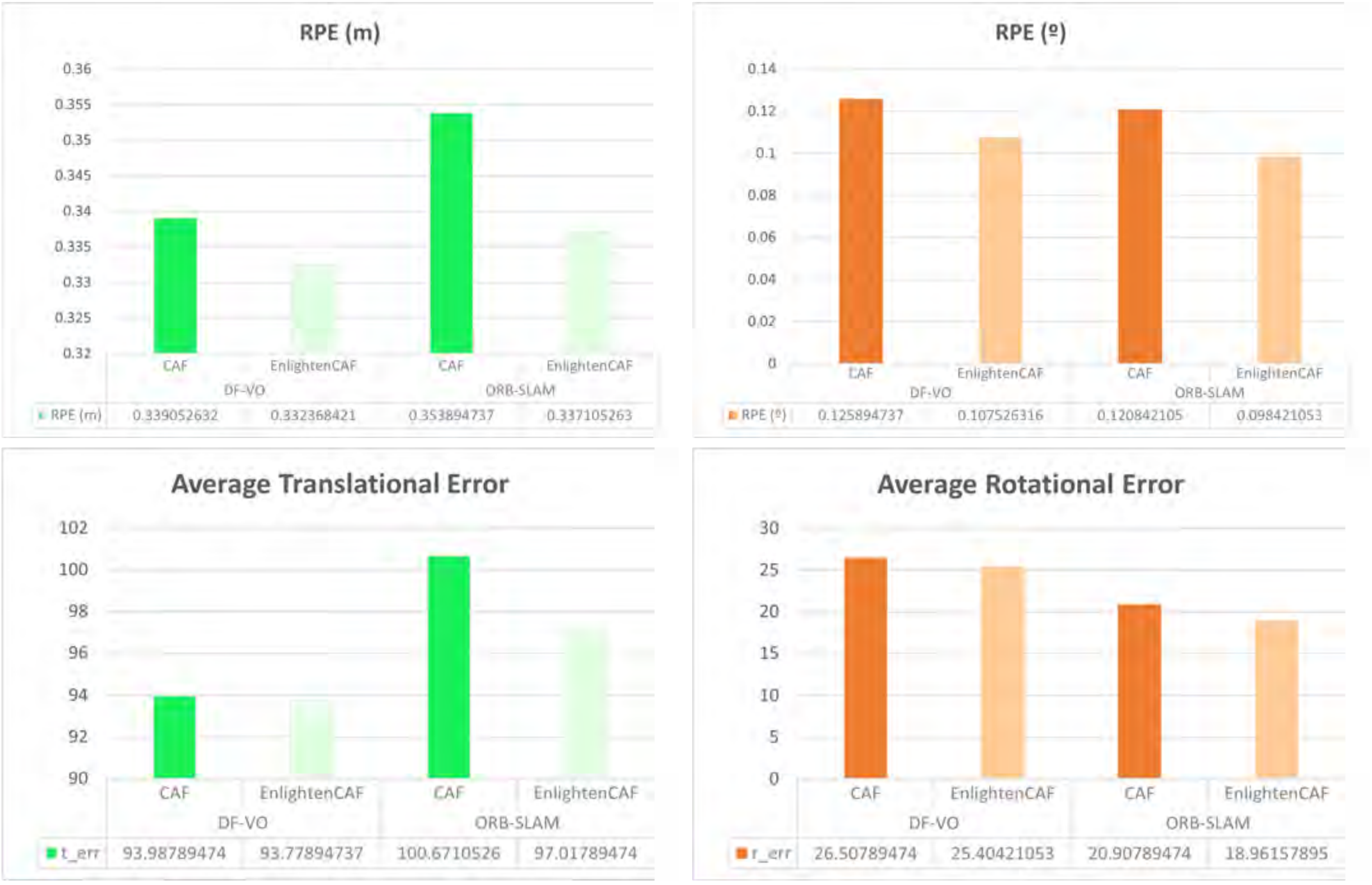


FIGURE 13: Comparison of relative VO evaluation metrics when applying DF-VO and ORB-SLAM2 algorithms in CAF and EnlightenCAF datasets. Translational and rotational components of relative errors are shown separately.

[14] Florian Tschopp et al. “Experimental Comparison of Visual-Aided Odometry Methods for Rail Vehicles”. In: *IEEE Robotics and Automation Letters* (2019), pp. 1–1. DOI: 10.1109/lra.2019.2897169. arXiv: arXiv:1904.00936v1.

[15] Volker Grabe et al. “Nonlinear ego-motion estimation from optical flow for online control of a quadrotor UAV”. In: *The International Journal of Robotics Research* 34.8 (2015), pp. 1114–1135.

[16] Yuliang Zou et al. “Learning monocular visual odometry via self-supervised long-term modeling”. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer. 2020, pp. 710–727.

[17] Huangying Zhan et al. “DF-VO: What Should Be Learnt for Visual Odometry?” In: *arXiv preprint arXiv:2103.00933* (2021).

[18] HAIDARA Gaoussou and PENG Dewei. “Evaluation of the Visual Odometry Methods for Semi-Dense Real-Time”. In: *Advanced Computing: An International Journal* 9.2 (2018), pp. 01–14. ISSN: 2229726X. DOI: 10.5121/acij.2018.9201.

[19] Rui Wang, Martin Schworer, and Daniel Cremers. “Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3903–3911.

[20] Hatem Alismail et al. “Direct visual odometry in low light using binary descriptors”. In: *IEEE Robotics and Automation Letters* 2.2 (2016), pp. 444–451.

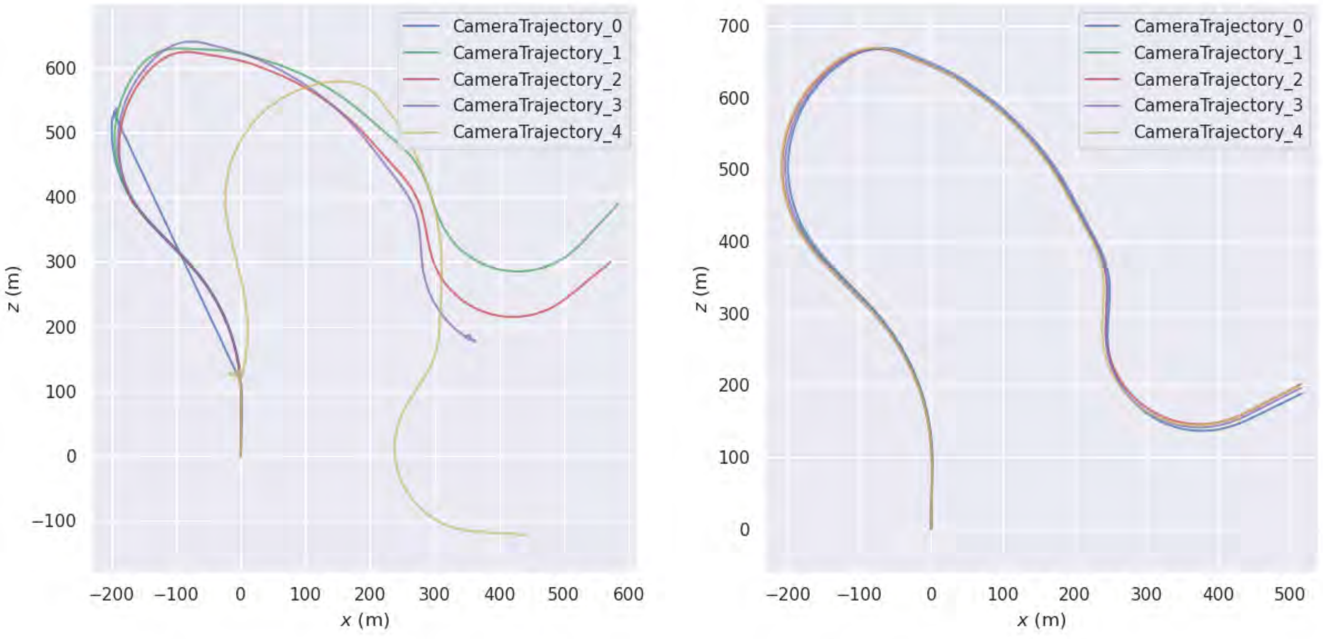
[21] Raul Mur-Artal and Juan D Tardós. “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras”. In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262.

[22] Ethan Rublee et al. “ORB: An efficient alternative to SIFT or SURF”. In: *2011 International conference on computer vision*. Ieee. 2011, pp. 2564–2571.

[23] Raúl Mur-Artal and Juan D Tardós. “Visual-inertial monocular SLAM with map reuse”. In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 796–803.

[24] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. “Visual SLAM algorithms: a survey from 2010 to 2016”. In: *IPSJ Transactions on Computer Vision and Applications* 9.1 (2017), pp. 1–11.

[25] Jeffrey Delmerico and Davide Scaramuzza. “A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots”. In: *2018 IEEE*



(a) *CAF* dataset

(b) *EnlightenCAF* dataset

FIGURE 14: Pose dispersion analysis on sample trajectory *01_54*. ORB-SLAM2 algorithm is executed five times on each dataset.

International Conference on Robotics and Automation (ICRA). IEEE. 2018, pp. 2502–2509.

[26] Berta Bescos et al. “DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes”. In: *IEEE Robotics and Automation Letters* 3.4 (2018), pp. 4076–4083.

[27] Chao Yu et al. “DS-SLAM: A semantic visual SLAM towards dynamic environments”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1168–1174.

[28] Ruben Gomez-Ojeda et al. “PL-SLAM: A stereo SLAM system through the combination of points and line segments”. In: *IEEE Transactions on Robotics* 35.3 (2019), pp. 734–746.

[29] Nan Yang et al. “Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 817–833.

[30] Huangying Zhan et al. “Visual odometry revisited: What should be learnt?” In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 4203–4210.

[31] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. “SVO: Fast semi-direct monocular visual odometry”. In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 15–22.

[32] Jakob Engel, Vladlen Koltun, and Daniel Cremers. “Direct sparse odometry”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.3 (2017), pp. 611–625.

[33] Lukas Koestler et al. “TANDEM: Tracking and Dense Mapping in Real-time using Deep Multi-view Stereo”. In: *Conference on Robot Learning*. PMLR. 2022, pp. 34–45.

[34] A. Kendall, M. Grimes, and R. Cipolla. “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2938–2946. DOI: 10.1001/jama.284.15.1980.

[35] Alex Kendall and Roberto Cipolla. “Geometric loss functions for camera pose regression with deep learning”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (2017), pp. 6555–6564. DOI: 10.1109/CVPR.2017.694.

[36] Sen Wang et al. “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2043–2050.

[37] Eric Brachmann et al. “Dsac-differentiable ransac for camera localization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6684–6692.

[38] Sang Jun Lee, Heeyoul Choi, and Sung Soo Hwang. “Real-time depth estimation using recurrent CNN with sparse depth cues for SLAM system”. In: *Interna-*

tional Journal of Control, Automation and Systems 18.1 (2020), pp. 206–216.

[39] Gabriele Costante and Thomas Alessandro Ciarfuglia. "LS-VO: Learning dense optical subspace for robust visual odometry estimation". In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1735–1742.

[40] B. Ummenhofer et al. "DeMoN: Depth and Motion Network for Learning Monocular Stereo". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 50.8–5047.

[41] Keisuke Tateno et al. "CNN-SLAM : Real-time dense monocular SLAM with learned depth prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 6243–6252.

[42] Huangying Zhan et al. "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 340–349.

[43] Nan Yang et al. "D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry". In: *arXiv preprint arXiv:2003.01060* (2020).

[44] Zhichao Yin and Jianping Shi. "Geonet: Unsupervised learning of dense depth, optical flow and camera pose". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1983–1992.

[45] Reza Mahjourian, Martin Wicke, and Anelia Angelova. "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5667–5675.

[46] Tianwei Shen et al. "Beyond photometric loss for self-supervised ego-motion estimation". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6359–6365.

[47] Mohammad OA Aqel et al. "Review of visual odometry: types, approaches, challenges, and applications". In: *SpringerPlus* 5.1 (2016), pp. 1–26.

[48] Yifan Jiang et al. "Enlightengan: Deep light enhancement without paired supervision". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 2340–2349.

[49] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[50] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[51] Duc-Tien Dang-Nguyen et al. "Raise: A raw images dataset for digital image forensics". In: *Proceedings of the 6th ACM multimedia systems conference*. 2015, pp. 219–224.

[52] Nima Khademi Kalantari, Ravi Ramamoorthi, et al. "Deep high dynamic range imaging of dynamic scenes." In: *ACM Trans. Graph.* 36.4 (2017), pp. 144–1.

[53] Jianrui Cai, Shuhang Gu, and Lei Zhang. "Learning a deep single image contrast enhancer from multi-exposure images". In: *IEEE Transactions on Image Processing* 27.4 (2018), pp. 2049–2062.

[54] Chen Wei et al. "Deep retinex decomposition for low-light enhancement". In: *arXiv preprint arXiv:1808.04560* (2018).

[55] Yuanfu Gong et al. "Enlighten-GAN for Super Resolution Reconstruction in Mid-Resolution Remote Sensing Images". In: *Remote Sensing* 13.6 (2021), p. 1104.

[56] Mahmoud Afifi et al. "Learning multi-scale photo exposure correction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 9157–9167.

[57] Zhangkai Ni et al. "Towards unsupervised deep image enhancement with generative adversarial network". In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9140–9151.

[58] Wei Xiong et al. "Unsupervised real-world low-light image enhancement with decoupled networks". In: *arXiv preprint arXiv:2005.02818* (2020).

[59] Ben Glocker et al. "Real-time RGB-D camera relocalization". In: *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2013, pp. 173–179.

[60] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. "Location recognition using prioritized feature matching". In: *European conference on computer vision*. Springer. 2010, pp. 791–804.

[61] A. Handa et al. "A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM". In: *IEEE Intl. Conf. on Robotics and Automation, ICRA*. Hong Kong, China, 2014.

[62] Santiago Cortés et al. "ADVIO: An Authentic Dataset for Visual-Inertial Odometry". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari et al. Cham: Springer International Publishing, 2018, pp. 425–440. ISBN: 978-3-030-01249-6.

[63] David Zuñiga-Noël et al. "The UMA-VI dataset: Visual–inertial odometry in low-textured and dynamic illumination environments". In: *The International Journal of Robotics Research* 39.9 (2020), pp. 1052–1060. DOI: 10.1177/0278364920938439. eprint: https://doi.org/10.1177/0278364920938439. URL: https://doi.org/10.1177/0278364920938439.

[64] Simone Ceriani et al. "Rawseeds ground truth collection systems for indoor self-localization and mapping." In: *Auton. Robots* 27.4 (Dec. 17, 2009), pp. 353–371. URL: http://dblp.uni-trier.de/db/journals/arobots/arobots27.html#CerianiFGMMMRST09.

[65] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. "Sun3d: A database of big spaces reconstructed using sfm and object labels". In: *Proceedings of the*

IEEE International Conference on Computer Vision. 2013, pp. 1625–1632.

[66] S Klenk et al. "TUM-VIE: The TUM Stereo Visual-Inertial Event Dataset". In: *International Conference on Intelligent Robots and Systems (IROS)*. 2021. arXiv: 2108.07329 [cs.CV].

[67] J. Sturm et al. "A Benchmark for the Evaluation of RGB-D SLAM Systems". In: *Proc. of the International Conference on Intelligent Robot Systems (IROS)*. 2012.

[68] J. Engel, V. Usenko, and D. Cremers. "A Photometrically Calibrated Benchmark For Monocular Visual Odometry". In: *arXiv:1607.02555*. 2016.

[69] F Walch et al. "Image-based localization using LSTMs for structured feature correlation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 627–637.

[70] Maurice Fallon et al. "The mit stata center dataset". In: *The International Journal of Robotics Research* 32.14 (2013), pp. 1695–1699.

[71] Hatem Alismail, Brett Browning, and M Bernardine Dias. "Evaluating pose estimation methods for stereo visual odometry on robots". In: *the 11th Int'l Conf. on Intelligent Autonomous Systems (IAS-11)*. Vol. 3. 2010, p. 2.

[72] Jürgen Sturm et al. "A benchmark for the evaluation of RGB-D SLAM systems". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 573–580.

[73] Thomas Schops et al. "A multi-view stereo benchmark with high-resolution images and multi-camera videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3260–3269.

[74] Nicholas Carlevaris-Bianco, Arash K. Ushani, and Ryan M. Eustice. "University of Michigan North Campus long-term vision and lidar dataset". In: *International Journal of Robotics Research* 35.9 (2015), pp. 1023–1035.

[75] José-Luis Blanco, Francisco-Angel Moreno, and Javier Gonzalez-Jimenez. "The Málaga Urban Dataset: High-rate Stereo and Lidars in a realistic urban scenario". In: *International Journal of Robotics Research* 33.2 (2014), pp. 207–214. DOI: 10.1177/0278364913507326. URL: http://www.mrpt.org/MalagaUrbanDataset.

[76] Will Maddern et al. "1 year, 1000 km: The Oxford RobotCar dataset". In: *The International Journal of Robotics Research* 36.1 (2017), pp. 3–15.

[77] Gaurav Pandey, James R McBride, and Ryan M Eustice. "Ford campus vision and lidar data set". In: *The International Journal of Robotics Research* 30.13 (2011), pp. 1543–1552.

[78] Jinyong Jeong et al. "Complex Urban Dataset with Multi-level Sensors from Highly Diverse Urban Environments". In: *International Journal of Robotics Research* 38.6 (2019), pp. 642–657.

[79] Daniel Olid, José M. Fácil, and Javier Civera. "Single-View Place Recognition under Seasonal Changes". In: *PPNIV Workshop at IROS 2018*. 2018.

[80] András L Majdik, Charles Till, and Davide Scaramuzza. "The Zurich urban micro aerial vehicle dataset". In: *The International Journal of Robotics Research* 36.3 (2017), pp. 269–273. DOI: 10.1177/0278364917702237. eprint: https://doi.org/10.1177/0278364917702237. URL: https://doi.org/10.1177/0278364917702237.

[81] Alex Zihao Zhu et al. "The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception". In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 2032–2039. ISSN: 2377-3774. DOI: 10.1109/lra.2018.2800793. URL: http://dx.doi.org/10.1109/LRA.2018.2800793.

[82] Tinghui Zhou et al. "Unsupervised learning of depth and ego-motion from video". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1851–1858.

[83] Michael T Ohradzansky et al. "Multi-agent autonomy: Advancements and challenges in subterranean exploration". In: *arXiv preprint arXiv:2110.04390* (2021).

[84] Haitao Zhao et al. "Development of a Coordinate Transformation method for direct georeferencing in map projection frames". In: *ISPRS journal of photogrammetry and remote sensing* 77 (2013), pp. 94–103.

[85] ÖPNVKarte map. *Planet dump retrieved from https://planet.osm.org*. 2017. URL: %5Curl % 7B % 20https://www.opnvkarte.de/#9.01;51.935;7%20% 7D.

[86] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. 2017. URL: %5Curl % 7B%20https://www.openstreetmap.org%20%7D.

[87] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[88] Yang Gao, Abdul Rehman, and Zhou Wang. "CW-SSIM based image classification". In: *2011 18th IEEE International Conference on Image Processing*. IEEE. 2011, pp. 1249–1252.

[89] Jacob Søgaard et al. "Applicability of existing objective metrics of perceptual quality for adaptive video streaming". In: *Electronic Imaging* 2016.13 (2016), pp. 1–7.

[90] Shinji Umeyama. "Least-squares estimation of transformation parameters between two point patterns". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 13.04 (1991), pp. 376–380.

[91] Michael Grupp. *evo: Python package for the evaluation of odometry and SLAM.* https://github.com/MichaelGrupp/evo. 2017.

[92] Huangying Zhan et al. *DF-VO*. https://github.com/Huangying-Zhan/DF-VO. 2021.

[93] Alexey Dosovitskiy et al. "Flownet: Learning optical flow with convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2758–2766.

[94] Charan Singh Rayat. "Measures of Dispersion". In: *Statistical Methods in Medical Research*. Springer, 2018, pp. 47–60.

MIKEL ETXEBERRIA-GARCIA received the B.S degree in computer engineering from University of the Basque Country(UPV/EHU), Donostia, Spain, in 2016 and the M.S. degree in advanced computer systems from the same university in 2018. He is currently pursuing the Ph.D. degree in applied engineering at Mondragon Unibertsitatea, Arrasate-Mondragón, Spain.

From 2016 to 2018, he was a system and database administrator in LKS, Donostia, Spain. Since 2018, he has been a Ph.D. student at Ikerlan Technology Research Centre, Arrasate-Mondragón, Spain. His research interest includes the application of Deep Learning techniques in railway domain, more specifically, on autonomous train navigation related tasks and on computer vision.

MAIDER ZAMALLOA received the Ph.D. degree in Physical Engineering from the University of the Basque Country(UPV/EHU), Bilbao, Spain, in 2010, in the field of pattern classification for speech technologies. She received the Computer Science Engineer degree from the University of the Basque Country (UPV/EHU), Donostia, Spain, in 2003. She is a certificied TUV Functional Safety Engineer for ISO 26262 (#13278/16).

She is a researcher at Ikerlan Technological Research Centre, Arrasate-Mondragon, Spain, since 2009. She is a senior researcher in the field of dependable embedded systems with +10 years experiences in testing of safety critical software for railway signalling systems (SIL4). Her research fields are dependable software and machine learning for computer vision.

NESTOR ARANA-AREXOLALEIBA did his PhD. at CNRS-LAAS (Robotics and AI Research Group). In 2020, he received the Ph.D. degree in Automatic System form the INSAT (Toulouse). From 2002 he has been working at Mondragon University. And from 2018-2019 he was a guest researcher in Robotics and Automation Research Group at Aalborg University (Denmark), where he was researching on reinforcement learning strategies for human-robot collaboration. Currently, he is working at Mondragon University in Robotics and Automation Research Group. He researches on Machine Learning and Image Processing for human-robot collaboration and autonomous vehicles. He has 50+ publications (Three books, two book chapters, five magazines and 40+ conference articles). Besides, he teaches on Robotics and Automation Master (ROS, AI-based Control and Mobile Robotics) https://www.researchgate.net/profile/Nestor_Arana-Arexolaleiba.

MIKEL LABAYEN studied Technical Telecommunication Engineering, specializing in image and sound at the Public University of Navarre (2005). He undertook his undergraduate dissertation at the Electronic Engineering department of the University of Surrey, U.K., in the audio and speech signal processing area. He completed his studies and he graduated from the faculty of Telecommunication Engineering in 2007, also at the Public University of Navarre. This time he undertook his master thesis at Digital Television and Multimedia Services department of the Vicomtech research center.

He started his professional carrier (2007-2012) as staff researcher in the field of computer vision - multimedia content analysis in the same research center. In 2012, he started a new career as co-founder and R&D project manager at Smowltech start-up (spin-off of Vicomtech) where he researches in the automatic facial and voice recognition area developing applications for online user authentication based on human biometrics. In the same period, he also was associate teacher at the Electronic Technology department of the University of the Basque Country. Currently, he is working in the field of intelligent transport systems (ITS) designing computer vision and machine learning based solutions for autonomous train operations in the railway sector, within the CAF group. He also is pursuing his PhD in computer vision and machine learning fields. His work as a researcher includes a number of publications and four patents.