COLLEGE MATHSTIC
DOCTORAL BRETAGNE
BRETAGNE OCEANE

ÉCOLE NATIONALE D'INGÉNIEURS DE BRST

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE
D'INGÉNIEURS DE BREST

ÉCOLE DOCTORALE N° 644
Mathématiques et Sciences et Technologies
de l'Information et de la Communication en Bretagne Océane
Spécialité : *Signal, Image, Vision, Son*

Par
Romain CAZORLA

Enhancing data informativeness in deep-learning based, point cloud semantic segmentation of industrial scenes

Thèse présentée et soutenue à Brest, le 26 Juin 2023
Unité de recherche : Lab-STICC, CNRS UMR 6285, team RAMBO
Thèse N° : 2023ENIB0002

Rapporteurs avant soutenance :

Dominique VAUFREYDAZ Maître de Conférences HDR, Université Grenoble Alpes
Rémi BOUTTEAU Professeur des Universités, Université de Rouen Normandie

Composition du Jury :

Président :	Benoit ZERR	Professeur de l'ENSTA Bretagne
Examinateurs :	Dominique VAUFREYDAZ	Maître de Conférences HDR, Université Grenoble Alpes
	Rémi BOUTTEAU	Professeur des Universités, Université de Rouen Normandie
	Antitza DANTCHEVA	Chargée de recherche HDR , Inria Méditerranée, Sophia Antipolis
	Benoit ZERR	Professeur de l'ENSTA Bretagne
Dir. de thèse :	Cédric BUCHE	Professeur des Universités, ENIB, CNRS IRL CROSSING
Encadrant :	Panagiotis PAPADAKIS	Maître de Conférences HDR, IMT Atlantique

Invité(s) :

Line POINEL Responsable Recherche et Innovation, Segula Technologies

Acknowledgement

I would like to first thank my thesis supervisors, Cedric Buche, Panagiotis Papadakis and Line Poinel, which allowed me to do this thesis. Their advices, knowledge and trust were what allowed me to grow as a PhD student. Thank you, Cédric, for the always timely, frank and insightful advices that allowed me to stay on the right track during those three years. Thank you, Panagiotis, for those discussions that led me to greater depths of reflexion. Thank you, Line, for your constant support, it was not possible to dream of a better manager during a PhD.

I would like to thank the members of my committee to have taken the time necessary to read this thesis and the thoughts they put behind their questions.

I am grateful to Segula Technologies for offering me the opportunity to do this thesis. The help of the company extended beyond financing this thesis by providing the necessary equipments, data and trainings. As a PhD student, being able to interact with the wide variety of research projects and personnel helped me in building my understanding of applied science.

I had the chance to work in a great environment thanks to my colleagues. Thank you, Emmanuel, Sara and Teddy for these conversations that allowed me to better understand other fields of researches. Some of your tips were helpful until the very end of this thesis. Microbiology seems less obscure of a science now. In the same regard, I also want to thank the naval team, especially Florian and Aloïs, for providing me the necessary occasional distractions to keep my mind sharp.

This thesis would have not be possible without the help of the draughtsmen and engineers, Edmond, Manu, Vincent, Xavier and Jean-Luc who took the time to give me the knowledge necessary to understand the available data. I would also like to thank Françoise Brunet, Perine Le Senechal, Nadine Conchez, Lucile Chaix, Vincent Baehrel, Vanessa Herry and Pauline Quere for their support during the thesis.

During those three years, I had the chance to supervise several internships that helped shape my work and the project surrounding it. I would like to thank Yuqi, Samy, Antoine, Noé, Pierre and Erell for their niceness and motivation. Those times were always rich in good moments and contributed a lot to the great environment at work.

None of this would have been possible without my family unconditional support. I have a special thought for Monique Cazorla, which could not see this thesis to its end.

Last, but not least, I would like to thank my spouse, Elise, and son, Noé, for their love which kept me happy and sane even in the hardest times.

Table of Contents

GLOSSARY

AIM Asset Information Model, an information model used during the operational phase of an asset life cycle*. 10

asset Something which has a potential or actual value to an organisation, for example a building or a piece of machinery*. 10

BIM Building Information Modelling, a process to create shared numerical models of an existing or to-be constructed physical asset. It is used to facilitate decision-making surrounding said asset during its whole life-cycle*. 9

CAD Computer Assisted Drawing is the appellation used to define software specialised in industrial drawing. Most of them can produce 2D drawings and 3D models and offer a range of tools facilitating the works of draughtsmen. 1

CAD model The data generated by a Computer Assisted Drawing software. Without context, it is understood as a 3D models generated by such software. 1

CDE Common Data Environment, a source of information for a given project or asset used by every actor needing to interact with an asset BIM*. 11

delivery phase Part of an asset life cycle where it is designed, constructed and delivered. It occurs before the asset can be used*. 10

federation Creation of a composite information model from separate information containers*. 11

information container Persistent set of information retrievable from within an organised system*. 11

information model Set of information container*. 11

life cycle The life of the asset, from its conception to the termination of its use, including development, usage, maintenance and decommissioning*. 10

operational phase Part of an asset life cycle where it is used and maintained. It occurs between its delivery phase and its disposal*. 10

PIM Project Information Model, an information model used during the delivery phase of an asset life cycle*. 10

RSS Random Surface Sampling, a method to transform mesh objects into point clouds by randomly creating points on the surfaces of a mesh. 66

SMARI Information Acquisition and Recognition Mobile System, "Système Mobile d'Acquisition et de Reconnaissance d'Information" in French, is the name of the project encompassing this thesis. It is also used as the name of the corresponding dataset. 2

trigger event Any event which modifies the asset and its associated information*. 10

VLS Virtual Laser Sampling, a method developed in this thesis to transform mesh objects into point clouds. It tries to simulate the acquisition process made by a real laser as well as the associated defect associated to such method of acquisition. x, 53

**Terms defined in the ISO-19650 Norm [1].*

List of Figures

All figures were created by the author except for Figure 2.5, supplied by Segula Technologies.

List of Tables

List of Algorithms

Synthèse en français — French Abstract

Cette thèse, située entre les domaines de l'apprentissage automatique et de la vision par ordinateur, s'intéresse à l'applicabilité de la segmentation sémantique par apprentissage profond de nuage de points quand ceux-ci représentent des environnements industriels. La thèse a été réalisée dans le cadre d'un partenariat entre l'École Nationale d'Ingénieurs de Brest (ENIB), l'IMT Atlantique, leur laboratoire associé, le Lab-STICC et l'entreprise Segula Technologies. Elle s'inscrit dans un projet plus vaste porté par Segula Technologies, SMARI (Système Mobile d'Acquisition et de Reconnaissance d'Information). Le but de ce projet est d'accélérer la vitesse à laquelle les dessinateurs projeteurs sont capables de recréer les plans et modèles CAO (modèles issus de logiciels de Conception Assistée par Ordinateur) d'une installation industrielle existante. Une étape essentielle de ce projet est de pouvoir segmenter sémantiquement des nuages de points représentant des installations industrielles ; c'est-à-dire, pouvoir associer à chaque point du nuage la fonction générale de l'objet dont il fait partie (tel que : tuyauterie, poutre, vanne...).

Grâce aux technologies actuelles, il est maintenant plus facile de reconstruire notre environnement sous forme de modèle 3D. Ces techniques facilitent certaines tâches communes, comme la visite de bien immobilier à distance. Dans le secteur industriel, l'utilisation de nuage de points 3D augmente chaque année et est codifiée par la norme ISO-19650 depuis 2018. Cette norme sur le processus de scan-to-BIM valide et structure les pratiques actuelles de numérisation industrielle. Les plans d'installations existantes sont souvent indisponibles ou obsolète et les projets de modification d'installation font souvent appel aux nuages de points afin de refaire ces plans. Au lieu de prendre des mesures sur sites pendant plusieurs jours, un laser peut être utilisé afin d'acquérir un nuage de points de l'installation. Malheureusement, traiter ces nuages de points afin de les transformer en modèles compréhensibles et par l'humain et par des logiciels de Conception Assistée par Ordinateur (CAO) est un processus laborieux. Une première manière d'améliorer ce procédé serait de donner la possibilité à la machine de comprendre le contenu d'un nuage de points 3D représentant une scène industrielle. Cette amélioration peut être associée à la capacité de réaliser une segmentation sémantique d'un tel nuage de points.

Si les avancées faites depuis 2017 sur la segmentation sémantique de nuage de points permettent la création de réseaux de neurones profonds performants, une impasse est faite sur l'application de ces méthodes au milieu industriel. Les méthodes d'apprentissage profond nécessitent une quantité importante de données et une base de données de taille

suffisante représentant des données industrielles est inexistante. Même en se restreignant à un seul type d'installation industrielle, la variabilité des milieux rend la tâche de création d'une base de données prohibitive. En créer une de taille comparable à celles de la littérature représente des centaines de jours de travail dû à la complexité des scènes impliquées. Cette thèse a donc dû se contenter d'une base de données restreinte. Cette difficulté a mené au problème principal traité par celle-ci : « Est-il possible d'améliorer les performances de segmentation sémantique en faisant une utilisation plus efficace des données d'entrainement ? ». Si les travaux réalisés par cette thèse se concentrent sur les données d'origine industrielle, les connaissances obtenues sont applicables à toute recherche s'intéressant à la segmentation sémantique de nuage de points. Deux contributions sont apportées afin de répondre à cette problématique d'utilisation des données.

La première, en étudiant la relation entre le réalisme des données synthétiques et les performances de segmentation sémantique. Pour cela, deux composantes sont considérées : la topologie et la couleur des données. La topologie d'une scène sous forme de nuage des points peut être évaluée sur trois niveaux : sa structure, la représentation de ses objets et finalement la structure locale de ses points. Cette thèse montre l'importance de la structure de la scène ainsi qu'une méthode d'échantillonnage permettant d'améliorer le réalisme de chaque objet synthétique lorsque celui-ci est représenté sous forme d'un nuage de points. L'efficacité de cette méthode d'échantillonnage est démontrée par une augmentation des capacités de segmentation sémantique d'un réseau entrainé sur des données synthétiques générées par cette méthode. La composante couleur est ensuite étudiée et une méthode permettant de colorer des données synthétiques est définie. Les cas où celle-ci améliore les performances de segmentation de manière significative sont ensuite présentés.

La seconde contribution est réalisée en se concentrant sur la manière dont les données sont fournies à l'entrée du réseau de segmentation. Plusieurs choses sont considérées : l'augmentation des données, leur découpe ou bien encore leur classification. Les travaux s'intéressent premièrement à un problème encore ouvert dans la littérature : la manière dont les données sont divisées. Les scènes étudiées sont trop grandes pour être fournies telles quelles au réseau de segmentation. Une étude de différentes possibilités de découpe, statiques et dynamiques est réalisée. Celle-ci montre la difficulté de réaliser une découpe dynamique dans le cadre des données industrielles mais propose plusieurs méthodes pour améliorer une découpe statique. Une découpe sous forme de piliers, se recoupant et étant fournis partiellement et aléatoirement à chaque epoch semble être le processus le plus efficace dans le cas des données industrielles. Deux augmentations de données sont ensuite étudiées : la rotation des données ainsi que l'ajout de la courbure du nuage de points dans les données. Appliquer une rotation autour de l'axe vertical est montré comme suffisant pour diminuer la faiblesse d'un réseau de segmentation par rapport aux rotations, même dans le cas de données industrielles où l'hypothèse de Manhattan tient fortement. En ce qui concerne la courbure, sa prise en compte ne permet pas d'améliorer les performances de

segmentation et l'utilisation des normales du nuage de points lui sera préférée. Finalement, la méthode de classification initiale des données est réévaluée. D'autres classifications sont considérées afin de mieux refléter l'utilisation des objets mais aussi des contraintes d'ordre pratique, tel que le déséquilibre inhérent aux données industrielles.

Chapter 1

Introduction

"They needed 800 hours to create this model from the point cloud, and it was only a surface modelling, not even a CAD model".

"When we began to use computer to draw, our team took more than five years to digitise all of our existing drawings".[1]

Modelling existing industrial facilities is tedious and time-consuming but often necessary. Multiple times during its lifetime, an industrial facility must be modified, which involves notably the creation of drawings. Those drawings are used to direct the work of construction teams as they contain installation instructions. Maintaining them up-to-date is also important for the day-to-day operation and maintenance work in the facility. They can for example be used as references when inspecting the facility, used for minor intervention or to diagnose dysfunction causes during operation. Ideally, those drawings should be constantly up-to-date and available to any party that needs them. This is not the case in practice, as drawings are lost, incorrect or preciously kept by engineering companies.

In our first collected statement, the facility in question was medium-sized (a building 140 meters long, 70 meters wide and 3 floors high) and the result was a surface model: a 3D reconstruction of objects composing the installation with no other information attached. Nowadays, in most industries, drawings and 3D models about industrial facilities are made with Computer Assisted Drawing (CAD) software and called, by extension, CAD models. Those models not only contain the 3D model of objects but also information about each of them. Thus, a pipe will be understood as a pipe, not a cylinder, and can be manipulated by specialised tools which ease the work of engineers and industrial draughtsmen. The surface model in the initial example had to be reworked for a significant amount of time to be integrated and used in a CAD software. For a larger industrial infrastructure, the time associated to recreate its piping and structural CAD model is estimated as 8900 man-hours by Agapaki [6].

Moreover, the industrial sector is currently subject to two external pressures, inducing the need for shorter modifications cycles. The global warming crisis, which asks for a rapid change in certain industrial practices and an increased demand for production flexibility, which drive the fourth industrial revolution. Those two external forces aggravate this old

1. Statements collected by the author during interviews with industrial draughstmen.

problem of balancing the different aspects surrounding industrial drawings: safety, cost and precision.

To create the model of an industrial facility, one must first go on-site to take measurements. Going to an industrial facility is, by itself, a source of risk (37.4% of injuries at work in the petrochemical sector were slip, trip and fall-related injuries in 2006 [54]). When done manually, this survey is a particularly time-consuming task. The use of a specialised measuring device reduces the time needed on-site, which diminishes risk but may represent a non-trivial investment for smaller companies. Older equipment included laser distance measuring devices and camera. Nowadays, lasergrammetry can also be used to acquire a point cloud of the facility. If this process is quicker and more precise than older techniques, industrial draughtsmen are still required to spend hours processing this point cloud, which is no better than a virtual visit of the facility. After all, point clouds only represent dense clusters of three-dimensional coordinates. They do not convey information on the equipment, purposes or risks associated to the scanned facility. Thus, be they working with manual measurements, pictures or point clouds, draughtsmen must still take the time to identify each object contained in a facility before modelling their respective position, orientation and geometry. Compared to the 90s, where the use of CAD software became popular and digitisation tasks similar to our second statement were occurring, CAD software user interface became far easier and quicker to use.

Some state-of-the-art software proposes tools such as automatic shape recognition of point clouds which reduces time spend on modelling. Time spent on piping can be reduced by 67% [6] for example. However, these tools are often cumbersome to use as they require an extended amount of user interaction. Multiple points of a single object must be selected before the software can propose a corresponding surface reconstruction. Multiple steps must still be carried out before a proper object model is created.

Looking at recent advances in point cloud understanding in the domains of autonomous driving or interior navigation, it should be possible to adapt those techniques to the industrial sector.

1.1 Context : the SMARI project

To reduce the difficulty in creating such drawings and finding enough industrial draughtsmen[2] SEGULA Technologies is carrying out a research project named SMARI ("Système Mobile d'Acquisition et de Reconnaissance d'Information") for Information Acquisition and Recognition Mobile System. The goal of the SMARI project is to automatize the most time-consuming parts of the drawing creation processes. This includes advancement

2. Three quarters of students in "BUT génie chimique - génie des procédés" prefer to continue their study before working. This number is obtained by looking at statistics published by three universities (available here, here and here). A similar assessment was made by recruitment experts and senior industrial draughtsmen: new hires are increasingly more difficult to find.

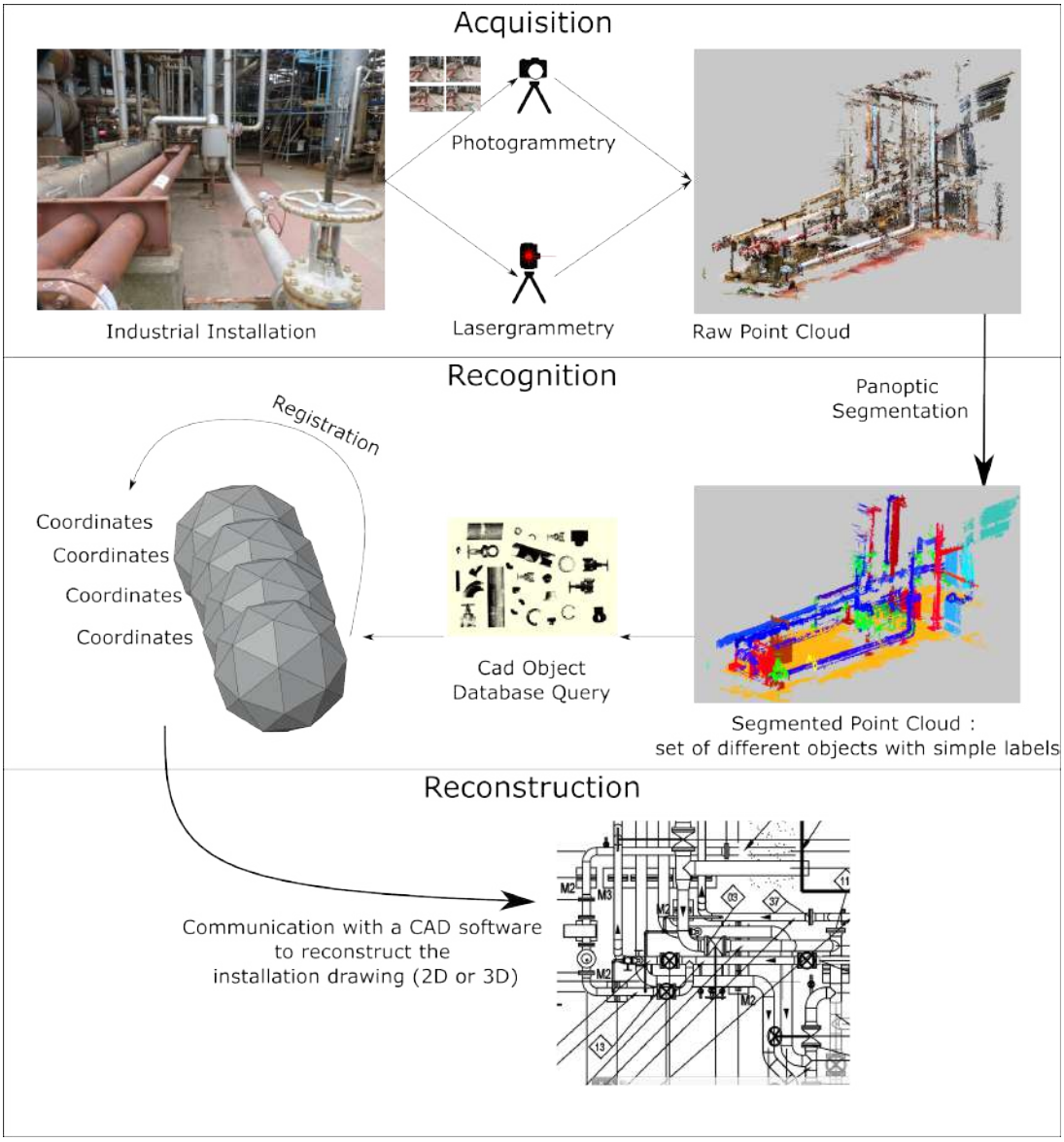


Figure 1.1: Complete devised process for the SMARI solution.

in acquisition techniques to drawing creation from identified CAD models. **This thesis is part of this project, focusing on the first recognition step : data segmentation.**

The envisioned solution is divided in three parts (Fig. 1.1 shows a detailed diagram of the solution). The first focuses on acquiring a point cloud of the industrial installation. The second part, recognition, takes this point cloud and tries to find, identify and position each object of interest in the point cloud. The last part of reconstruction defines communication protocols with CAD software to create drawings usable by the draughtsmen.

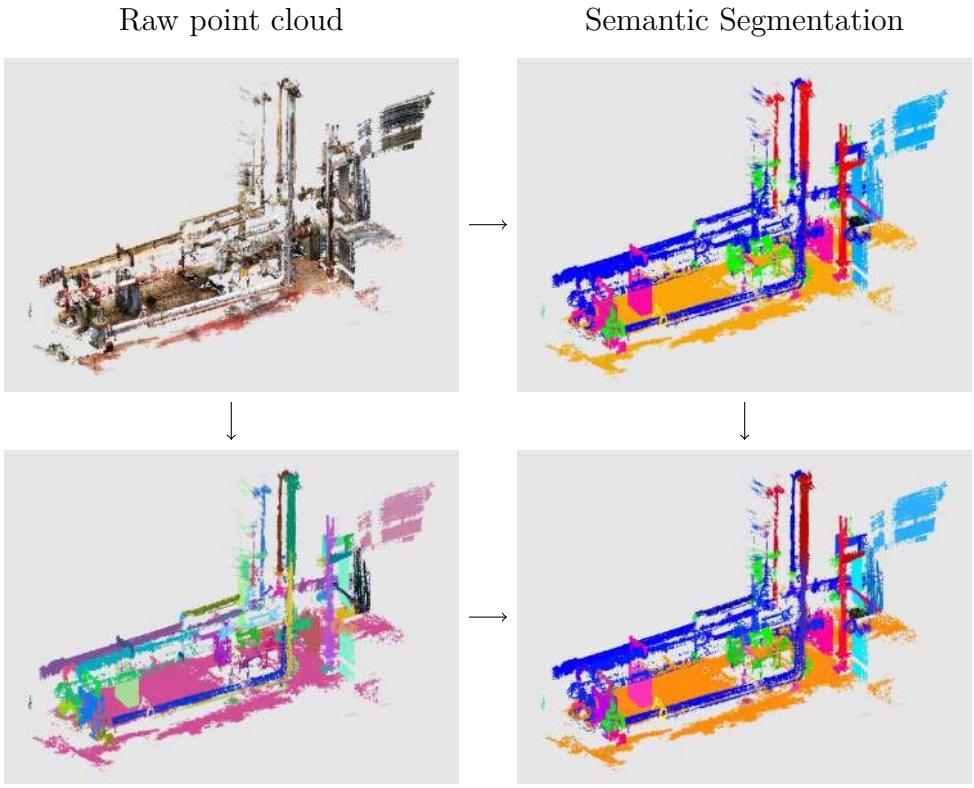


Figure 1.2: Illustration of different ways to segment a point cloud.

1.2 Point cloud semantic segmentation of an industrial scene

To perform recognition, two main techniques are possible: a detection-based technique and a segmentation-based one. Both could be used to obtain a first knowledge of the point cloud, find objects of interest and give enough information for further processing of these objects. Segmentation was chosen as almost every object in acquired point clouds is of interest[3].

Initially, the goal of this thesis work was thus to find a method to segment a point cloud representing an industrial infrastructure. Semantic segmentation was chosen as a means to achieve it because it associates semantics to each point. However, this does not suffice to identify different objects of the same semantic class. Adding an instance segmentation component to the process will be accomplished after the thesis, during the realisation of a first SMARI prototype. This hybrid task is called panoptic segmentation. Examples of segmentation, obtained manually, are presented in Fig. 1.2.

3. Less than 2 % of acquired points in our database represent objects which should not be modelled or are too unique to be considered.

1.3 Constraints and objectives for the semantic segmentation

As this thesis is integrated in a larger project, some conditions must be respected to ensure its usefulness. Those constraints are on segmentation quality, resilience to input degradation and processing time.

The bulk of industrial scenes is composed of simple objects. Those are mainly pipes and structural elements which also correspond to the objects which take the most time to model: as they are of varying length, measurements must be taken manually before drawing. Achieving a high segmentation precision on those classes is a priority. Electrical elements are considered but of lower priority. This disinterest is mostly due to the current quality of point cloud scans that can only poorly capture electric cables and the lower gain in time possible by automatizing this task.

The method must be robust to variation and degradation on the input data. The sensor used to acquire data is susceptible to change depending on the available resources and on-site constraints. Point clouds obtained by lasergrammetry and photogrammetry are valid inputs but they generate point clouds whose local point structures are different. Lasergrammetry creates more organised point clouds reflecting the sweeping movement of the laser. Point clouds from photogrammetry contain points which are less organised, as they originate from features of interest in the acquired images. The possible change of acquisition hardware also means an important variation on point density, position noise and dropout noise. Furthermore, acquisition conditions and the disposition of the scene could create important occlusions and point density variation in the final point cloud. Acquired objects can also present imperfection, such as rust, paint or mark which impact colour, surface orientation and position information. Finally, project requirements necessitate robustness to rotation along the Z-axis (corresponding to gravity) and translation of the scene: the scene could be placed in a larger setting, far from the scene centre of the used coordinates system.

Lastly, the devised method must be able to segment a scene during the operator downtime. The envisioned downtime are lunch, night and weekend, depending on acquisition context and scene size. The envisioned scenarios are:

- Scanning a small part of an industrial setting in the morning, which could be processed during lunchtime.
- Launching the SMARI process for the night after spending the day on-site.
- Spending several days on-site to acquire a medium to big installation and having the data processed during the weekend.

A first prototype of the global project being absent, it is difficult to determine the time

allocated to the segmentation step. An inference speed in the order of the second per square meter is taken as constraint. This allows to process a 30 m by 30 m scene in 15 minutes by the segmentation method and should let enough time to the global project to transform this scene into a CAD model.

1.4 Research questions

How can we semantically segment a point cloud representing an industrial scene? From this short and seemingly simple question, stem several other ones. Based on the existing literature, the use of deep learning is an obvious answer. However, a major constraint exists in the absence of publicly available datasets geared towards point cloud semantic segmentation in an industrial setting. The time needed to construct a sufficiently diverse dataset to represent even a fraction of possible industrial settings (even when restrained to oil & gas installations only) is prohibitive. Wherever the studied scenes are sparse in their organisation (such as an urban setting for autonomous driving, where most objects are convex and, hopefully, do not contain each other), billions of points are considered[14]; this would represent more than 400 man-days of work with industrial scenes of medium complexity. As industrial scene content is complex, there is a high variability in industrial object sizes and shapes and each object's semantic classification depends on their context (e.g. a cylinder can be either a pipe or some structural element), a higher quantity of data would be required so that it would be sufficiently representative.

When working with a restrained quantity of data, several solutions exist. Transfer learning and few-shot learning have shown their advantages in the literature but they do not deal with the domain-specific constraints (scene complexity, object variability and context, object shape classification). Alternatives comprise data modification techniques, either by handling and transforming the data in various ways (data augmentation) or by adding synthetic data (synthetic data augmentation). This reasoning leads to the general research question (**GRQ**): **"Can point cloud semantic segmentation performance be increased by a more effective use of training data?"** We will focus our interest in this question to the oil & gas application domain.

This general research question can be divided in two when looking at synthetic data and data augmentation.

Research Question 1 (RQ1): **Can synthetic data be processed in a way that reduces domain gap?** Synthetic data augmentation seems to be the most pertinent and obvious solution (in view of increasing data quantity) to the lack of data problem. Reliable data is readily available in the form of Conception Assisted Drawing (CAD) models from specialised software, which helps in solving not only the lack of data problem but also other domain specific problems. However, synthetic data constitutes an idealised or synthetic version of real world data, which creates a domain gap that impedes the learning process

of neural networks. Such networks can be transformed and trained to be able to cope with this gap. Nonetheless, bridging this gap beforehand can be thought as a sensible approach which leads us to our first research question. Answering this question also compels us to look at the quality of synthetic data. Does using synthetic data that are made more realistic improve segmentation performance? If so, which point cloud characteristic does a semantic segmentation network consider more important?

Research Question 2 (RQ2): **Does the way data is presented to the segmentation network influence its performance?** Data augmentation is often used to increase the informativeness of training data. The use of these techniques could help us in solving some problems inherent to industrial data and constraints due to the SMARI project. Extending the concept of data augmentation to operations applied to data before the segmentation network input layer bring us to the second research question. For the purpose of this research question, we define these operations as Data Representation Operations. Thus, considered data representation operations can be usual data augmentations, but also be the division process which most state-of-the-art work applies to data or the labelling step inherent in creating a dataset.

1.5 Contributions

To answer our General Research Question, a new data processing workflow was devised. This workflow can be divided in two parts, each answering **RQ1** and **RQ2** respectively.

RQ1: Contributions towards answering this question include studying the relationship between synthetic data realism and point cloud semantic segmentation performance. For this task, two components of realism are considered. First, the topology of the scene can be thought about at three scopes: global scene structure, object representation and local structure. The importance of the synthetic scene structure for the network context understanding capability is shown. A new virtual laser sampling method is presented. This sampling method improves each synthetic object realism as well as the points local structure and leads to an increase in segmentation performance. The second component of realism is colour. The influence of colour on segmentation is studied and a synthetic data colouring method that improves segmentation performance in some cases is proposed. A hypothesis towards the origin of failure cases in the colouring method abilities is also advanced.

RQ2: Contributions were made by focusing on different data transformations applied before the segmentation network input layer. The way by which these transformations influence segmentation performance and ways to improve them are studied. These steps are divided in three categories: data augmentation, data splitting and dataset labelling. Two data augmentations are studied: rotation along the vertical axis and pre-computing the point cloud curvature instead of using points normal as a feature. The rotation

augmentation is shown to alleviate greatly a segmentation network weakness towards rotation, even in an industrial scene where the Manhattan hypothesis holds strongly. On the other hand, curvature as a point cloud feature does not bring any segmentation improvement. As an industrial scene covers large areas, different splitting strategies are studied. The density of the input data is found to be more important for the segmentation network than the size of the area presented. In dividing the scene, a pillar primitive is more successful than other primitives are. The benefits of variability in the division process are also showed. Finally, initial data labelling is reassessed. Other class partitioning options are considered to better reflect object use but also practical constraints in the dataset such as class imbalance.

1.6 Organisation of the manuscript

The remainder of this manuscript is divided as follows:

Chapter 2 first shows today's use of data in the industrial context, notably by briefly presenting the recent ISO-19650 norm. Works on point cloud semantic segmentation methods linked to the industrial sector context are then presented. In its second part, this chapter defines more rigorously the task of point cloud semantic segmentation and presents currents state-of-the-art methods. These methods are then studied under the view of data use efficiency. Finally, recent methods to generate and modify synthetic data are presented in this chapter.

The objectives defined in this introduction as well as the specificities and challenges associated to working with point clouds representing industrial scenes are detailed in Chapter 3. Following the assessments made in its first part, this chapter then presents an overview of our data processing workflow, a proposition towards solving **GRQ**. The experimental methodology used in most experiments is described at the end of this chapter.

Our contributions towards answering **RQ1** are detailed in Chapter 4. The method is first presented before experiments are carried out. The factors of realism, scene topology and data colour are studied separately. The sampling method is validated through the results of the experiments on synthetic data topology. Success and failure cases of the colouring method are presented, with strong arguments on the cause of its failure.

Our contributions towards answering **RQ2** are detailed in Chapter 5 where sections describing the methods and experiments for each category of data augmentation, data splitting and dataset labelling are included.

A validation of our conceived data processing workflow is made in Chapter 6.

Chapter 7 presents in its first part the conclusion on the contributions done toward answering our general research question. Future works to expand on these contributions are presented in its second part.

Chapter 2

Background

This background chapter contains an overview of the industrial methods for which point cloud semantic segmentation could be applied. The norm ISO-19650 is first described, before recent works on deep learning applied to the industrial sector are presented. The general problem of point cloud semantic segmentation is then posed before related works are exposed. This highlights two gaps in the current literature: the lack of techniques to reduce domain shift for synthetic data augmentation when point clouds are concerned, as well as a lack of research on the influence of point cloud pre-processing towards the ability of deep neural networks to perform PCSS.

In this chapter, background knowledge on the usefulness of point clouds and their comprehension for the industrial sector is first presented in Section 2.1. Once the current industrial methodologies and research in this sector are presented, Point Cloud Semantic Segmentation (PCSS) can be studied more generally. This study begins by presenting formal definitions of the concepts (point cloud, PCSS, some metrics) used in this thesis in Sec. 2.2. It is followed by a presentation of related works on PCSS (Sec. 2.3) before a conclusion on this chapter can be drawn in Section 2.4.

2.1 Industrial context

Point clouds use in the industrial sector is mostly destined to be a tool helping CAD modelling. The use of point cloud acquisition techniques (e.g., lasergrammetry or photogrammetry) substitutes more manual measurement techniques and reduces the time spent on site by industrial draughtsmen teams, increasing their safety. The use of such techniques is increasing in the industry and is mostly known as "Scan-to-BIM". Scan being the acquisition process of the scene, with the second part corresponding to the use of this point cloud to create a BIM (Building Information Model) of the scene. In order to set guidelines for these new techniques, the ISO norm 19650 was first published in 2018, with

subsequent additions and modifications until September 2022. This norm is first briefly explained in Section 2.1.1, with a focus on parts where point cloud semantic segmentation could ease the application of this norm. A second part, Section 2.1.2, focuses on the few existing works connected to the use of point cloud semantic segmentation applied to the industrial sector.

2.1.1 Norm ISO-19650

The ISO-19650 norm, entitled "Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM)", concerns the creation, use, share and life of BIM information during and after an asset life cycle (conception, creation, operation, destruction). It is divided in five parts:

1. Concepts and principles [1]
2. Delivery phase of the assets [2]
3. Operational phase of the assets [3]
4. Information exchange [4]
5. Security-minded approach to information management [5]

When this norm is well-followed, it leads to the creation of two models, the Project Information Model (PIM) and the Asset Information Model (AIM). The first one is used during the delivery phase (part 2) and the second one during the operational phase (part 3) of the assets. When taking the example of a building construction project, the PIM consists of the different drawings used during the construction. The AIM is the model of the building once it is finished. Those models can contain the CAD modelling of the building but are not limited to this kind of information. Additional information related to the project can be considered such as maintenance cost, scheduling or maintenance dates. This information must be updated following a defined schedule and level of detail, as described in part 4 of the norm.

When considering point cloud semantic segmentation and a project which uses this technique, such as the SMARI project whose this thesis is part of, several aspects of the ISO 19650 norm can be facilitated. For a complete BIM project as described by the norm (as illustrated in Figure 2.1), three axes exist:

- At the beginning of the project, when creating the initial PIM and existing data is insufficient.
- At the end of the project, to verify the virtual model against the real asset. This increases the reliability of information when going from a PIM to an AIM.
- At each trigger event during the operational life of the real asset, to verify the trustworthiness of the existing AIM.

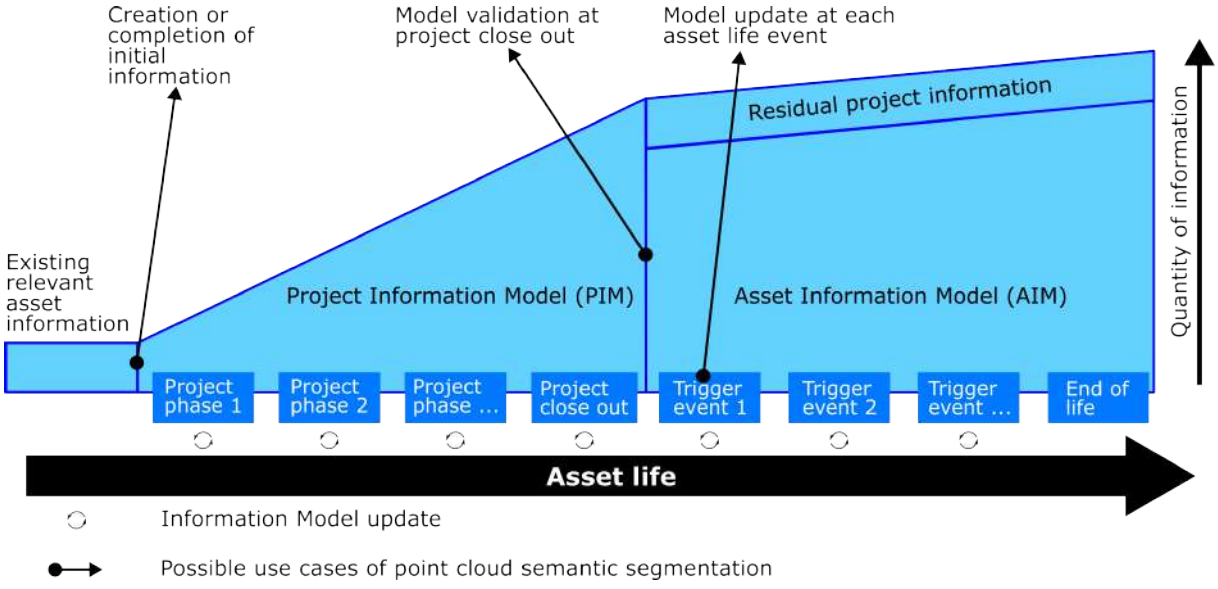


Figure 2.1: BIM information modelling during an asset life as described in the ISO 19650 norm.

ISO 19650 defines three levels of maturity for information management. The first level corresponds to working on unfederated information models. In this level, each information related to an asset are not organised together. The next step is the federation of information model with a mix of manual and automated information management processes. The last step of maturity demands for an enhancement in the information exchange process with change in the Common Data Environment (CDE). In the first two stages, the information containers are contained in the information model. In the third stage, an information container can be decentralised and queried from a database. However, going from a File/Model/Container CDE towards a Query/Model/Container CDE demands for standards which are not yet created.

Thus, a good use of point cloud semantic segmentation could substantially increase the adoption of the latest scan-to-BIM norm and further help in maturing analogue and digital information management by automating some processes. A major constraint when looking ahead would be the flexibility needed to work with different Common Data Environments. If such problem is considered by a point cloud semantic segmentation method, it will need to be able to work on fewer data than classical deep learning methods and only considers classes of objects which are sufficiently general.

2.1.2 PCSS applications to the industrial domain

The previous section showed both the utility and relevance of point cloud semantic segmentation for the current work process used in the industrial domain. However, few works focus on this kind of application in the scientific literature.

The closest work to our goal is the recent PhD thesis of Agapaki [6] published in 2020. This thesis focuses on the segmentation of oil and gas industrial scenes. An analysis of the most common objects present in this domain is made, followed by a time-cost analysis of modelling industrial scenes from point cloud by using a state-of-the-art software. This analysis identifies pipes and structural elements as those taking the most time to model, mostly due to their high frequency of appearance in such environment. Following those results, Agapaki proposes a method to detect the simple shapes which describe those elements. For the work of Agapaki, such method is sufficient as it can be integrated directly into a modelling software to improve some of its tools. Finally, it proposes a dataset of oil and gas scenes, CLOI, which is unfortunately unavailable as of now.

Except for CLOI, other datasets applied to the industrial domain exist. The EIF dataset [107][50] proposes point clouds representing individual industrial piping components such as valves, pipes, tee, reducer or flange. Those point clouds are acquired from real scenes and are designed to be used in retrieval tasks. Lastly, Yan et al. [108] proposes an incomplete semantic segmentation dataset, PSNet5, composed of beams, pipes, pumps and tanks only. They achieve good segmentation performance on this dataset by combining deep residual learning with PointNet++ [76], a classical point cloud semantic segmentation method[1].

2.2 Formal definitions

For the sake of rigour and completeness, formal definitions are first provided in Sec. 2.2.1. These definitions will be followed by the description of the metrics used in this manuscript to determine segmentation quality (Sec. 2.2.2), notably in the following related work section (Sec. 2.3).

2.2.1 Point Cloud Semantic Segmentation

As data are at the centre of any machine learning process, we will first define our data of choice: point cloud.

***Definition* 1** (Point cloud)**.** A point cloud is defined as a set of points P of size $N \in \mathbb{N}$ with each point defined as $p_i \in P\,, \forall i \in [\,1; N\,]$. A point p_i of P is a coordinates vector of dimension $\mathfrak{D} \in \mathbb{N}$. □

Colloquially, P is said to be of dimension $\mathfrak{D}$. In this thesis, we only consider the case where the points are in a 3D space (i.e. $\mathfrak{D} = 3$ and thus $p_i \in \mathbb{R}^3$). This first definition represents a standalone point cloud: only point coordinates are known. By convention, we will use x, y and z as the names for the axis of the metrical 3D space where point clouds

1. Details on point cloud semantic segmentation methods are provided in Section 2.3

are represented. The coordinate frame of each point cloud is aligned with gravity. The vertical axis is z and an increase of coordinate in z corresponds to an upward direction. As seen in Section 1.3, the origin of the coordinate frame is not necessarily present in the cloud.

***Definition* 2** (Features of a point cloud)**.** Let P be associated to a set of features. Let F, a set of cardinality N, be the features of the point cloud P. Each feature $f_i\,, \forall i \in [\,1; N\,]$, of F is defined as a vector of size $\mathfrak{F} \in \mathbb{N}$. f_i is called the feature of the point p_i for all points of P. □

Such a feature can be for example the point colour or normal vector[2] but also features as computed by a convolutional neural network. We consider the case where features can be represented as real numbers (i.e $f_i \in \mathbb{R}^{\mathfrak{F}}$).

Finally, the order of the points in a point cloud is arbitrary. It is then possible to define an equivalence of two point clouds as such:

***Definition* 3** (Point cloud equivalence)**.** Two sets of points $P_1\, P_2$ are considered equivalent if their cardinality and their points are equal : $|P_1| = |P_2|$ and $\forall\ p_{1_i} \in P_1 \mid i \in [\,1; |P_1|\,]\ \exists\ p_{2_j} \in P_2 \mid j \in [\,1; |P_2|\,]$ such that $p_{1_i} = p_{2_j}$. □

A more strict equality between two point clouds can be constructed if the features of each point are also considered ($p_{1_i} = p_{2_j}$ and $f_{1_i} = f_{2_j}$). This stricter definition is equivalent to defining point cloud equality as a set equality.

***Definition* 4** (Semantic Segmentation)**.** Let us consider a point cloud P of size N. For each of its points p_i exists a semantic class $c_i \in \vec{C} \mid \vec{C} \in \mathbb{N}^N$ with $i \in [\,1; N\,]$. The goal of semantic segmentation is to find a mapping M such that $M(P) = \vec{C}$. The set of all possible class values is defined as $C \in \mathbb{N}$. □

2.2.2 Metrics

To quantify the segmentation quality performed by each method, some metrics should be defined. The following metrics are defined when possible in terms of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

***Definition* 5** (Accuracy)**.**

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

□

2. Those features are typically input features of a neural network. The normal vector in particular must be computed using the neighbourhood of each point to approximate a local surface from which it can be estimated.

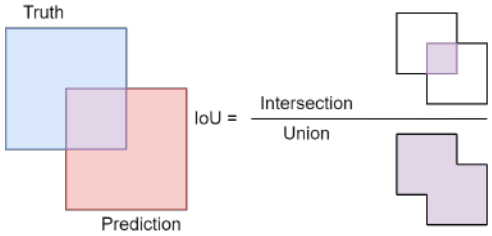


Figure 2.2: Graphical illustration behind the IoU metric.

***Definition* 6** (Precision)**.** The average precision (AP) over all data.

$$AP = \frac{TP}{TP + FP} \tag{2.2}$$

When considering by class precision for semantic segmentation, it is possible to define a mean average precision (mAP) : for C, c_i and i as in Def. 4, $\exists AP_i$ such that :

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C} \tag{2.3}$$

□

***Definition* 7** (Intersection over Union)**.** The intersection over Union (IoU), sometimes called Jaccard Index, represents the correctly guessed points over mislabelled points (see Fig. 2.2). Contrary to the precision, it is almost exclusively used to evaluate semantic segmentation. IoU is defined by class, with the mean (mIoU) used to aggregate the result over classes.

$$IoU_i = \frac{TP}{TP + FP + FN} \tag{2.4}$$

$$mIoU = \frac{\sum_{i=1}^{C} IoU_i}{C} \tag{2.5}$$

□

Finally, computation time can also be used as a metric: it can be expressed in terms of points or scene surface inferred per unit of time.

2.3 Related Work

As we saw before, point cloud semantic segmentation is the act of associating a class to each point in a cloud (Sec. 2.2). There are three ways to perform this task [104].

The more conventional methods are based on optimisation techniques that segment the point cloud. Once this primary segmentation is done, it is then possible to associate a semantic class to each point or group of points.

A second category of techniques is based on traditional machine learning. The point cloud is first decomposed in neighbourhoods for which features are computed. Supervised classification techniques are then applied to these neighbourhoods, such as Support Vector Machine or neural network, to compute the class of each group of points.

Finally, there are the more recent deep learning based methods. Contrary to traditional neural networks techniques, deep learning possesses several peculiarities [71]:

- It necessitates an important increase in computational power in order to train the network. The advent of GPU-based computation is one aspect behind the current success of deep learning methods.
- It necessitates a tremendous quantity of data in order to correctly train the neural network.
- It possesses the ability to extract the features necessary to train the decision part of the network by itself. This is one reason behind its wide adoption as it removes the laborious work of finding the best descriptors associated to each specific task.
- It relies on increasingly larger neural network models. This last characteristic is the reason behind its name: a deep neural network can be dozens or hundreds of layers deep.

Those methods originate from Yann LeCun works in the 90s [56] and were popularised in 2012 by AlexNet [53] when sufficient computational power and sufficiently large dataset [32] were made available. In their cases and in the case which are of interest for our problem, convolutional layers are used by the network to extract interesting features from the data.

With both the original method of Yann LeCun [56] and AlexNet [53], the medium of application are 2D images. Images are advantageous in being composed of regular grids of pixels, which eases the implementation of discrete convolution. However, point clouds are not represented as a regular structure but as a list of points residing in a continuous 3D space. This difference in data representation requires using specific techniques in order to apply deep learning to 3D point clouds and can be summarised by Figure 2.3. The different families of methods are presented in Section 2.3.1.

The first task where deep learning was applied to is classification, which consists in associating a unique class with a data instance, such as an image. Semantic segmentation is a more difficult task which focuses on understanding each component of the data studied. This difference leads to a difference in the general network architecture used, which is presented in Section 2.3.2.

Once those two areas of knowledges necessary to understand point cloud semantic segmentation are presented, the state of the art can be studied. First the datasets commonly

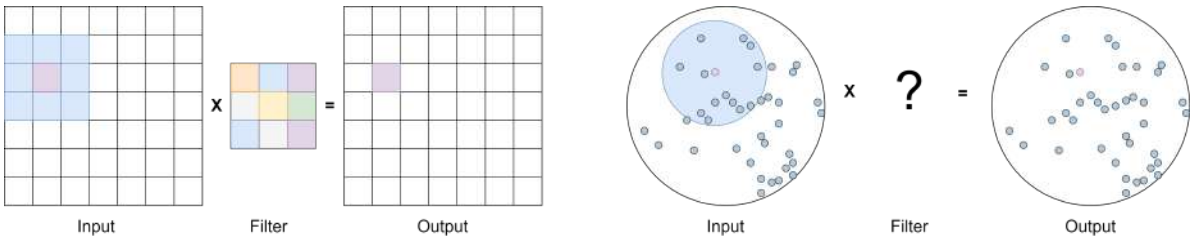


Figure 2.3: Illustration of the main problem behind applying deep learning techniques to point clouds: the non-regularity of point clouds.

used in the literature are described in Section 2.3.3.1 before comparing the different point cloud semantic segmentation methods in Section 2.3.3.2. In subsequent Section 2.3.3.3, a possible problem with current networks is raised: little research is done on the influence of the PCSS network input on its performance.

As seen previously in Section 2.1.2, the few datasets available for point cloud semantic segmentation of industrial scenes are either incomplete or unavailable. To alleviate this problem, synthetic data augmentation is studied in Section 2.3.4.

2.3.1 Deep learning with point cloud

In most works, deep learning methods applied to point clouds are divided in three families of methods depending on how they cope with the non-regular structure of point clouds [104][38]:

- The Multi-view family of methods tries to circumvent the problem by transforming the point cloud into a set of images. After this transformation, those techniques can rely on advances made in 2D deep learning to process the obtained images.
- The voxel family of methods contains those which transform the irregular structure of the point cloud into a 3D grid of voxels. As the obtained structure is regular, it becomes trivial to transfer advances on 2D pixel grid to 3D voxel grid.
- The point-based family of methods works directly on the point cloud with methods able to extract characteristics by themselves from irregular structures.

This last family of techniques appeared in 2017 from the pioneer work of PointNet [75]. Since then, most methods of deep learning applied to point clouds try to work directly on the cloud and belong to this third category of techniques. Following this observation, the categorisation of Guo et al. [40] feels more appropriate to reflect the current state-of-the-art. In this categorisation, presented in Figure 2.4, only two general families of methods exist: projection methods and point-based methods. Projection methods are those that modify the structure of the point cloud to use advances on deep learning made on other data structures. Point-based methods are those working directly on the cloud and diversified in three sub-categories since PointNet.

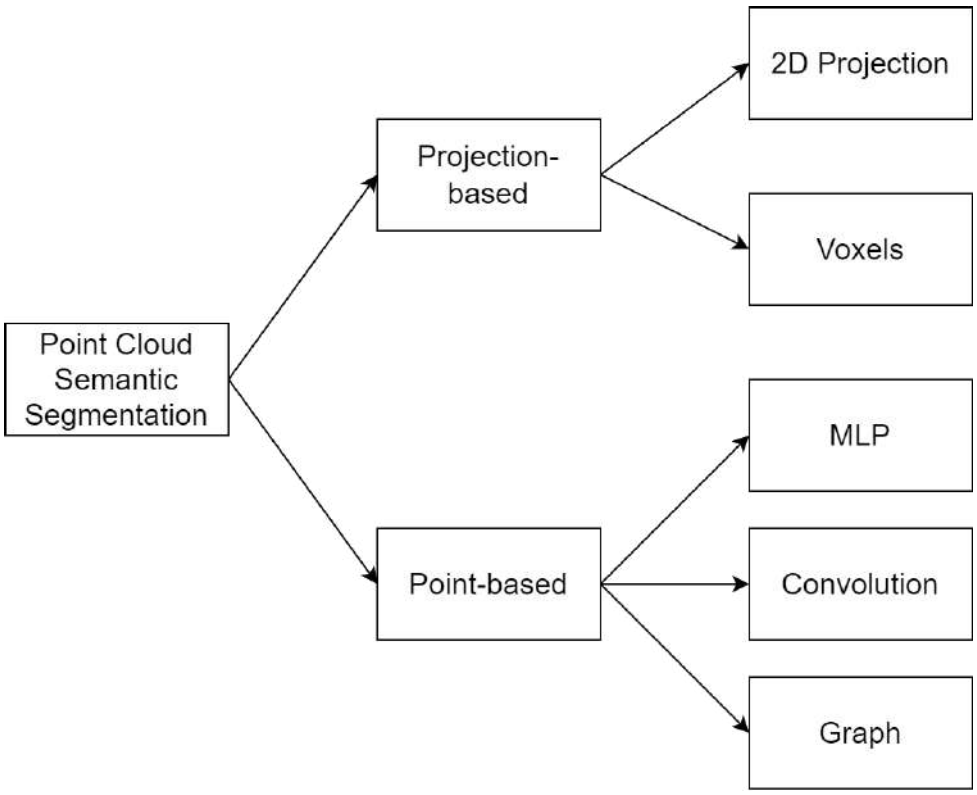


Figure 2.4: Classification of deep learning methods applied to point clouds, adapted from the work of Guo et al. [40].

2.3.1.1 Projection-based methods

Projection based methods were the first to successfully apply deep learning to point cloud and offered good results on task such as classification [74]. They all possess the advantage of transforming a hard problem (applying deep learning to irregular structure) into an easier problem (transforming the point cloud in a structure where applying deep learning is a solved task). When looking at tasks such as classification, where each point cloud is considered as a whole, the projection does not need to be bijective. However, for tasks where each component of the data must be understood, such as semantic segmentation, a back-projection is necessary. In those cases, the projection-based methods are unequal in their abilities. Examples of 2D projection and voxels projection of an industrial object are presented in Figure 2.5.

Voxels methods partially conserve spatial information between points and can be seen as working on low-resolution points clouds. As a 3D space can contain empty space, it is much less efficient than 2D images where every pixel contains information. The cost of increasing the voxel resolution is also higher, as it evolves cubically instead of quadratically. To solve these problems, Maturana and Scherer used an occupancy grid [64] and succeeded in working in real-time for low-resolution voxel grids (32 x 32 x 32). Another approach is to work with octree [94][77]. Those octrees allow computing convolution on voxels of different sizes and partially alleviate the resolution problem. To back-project

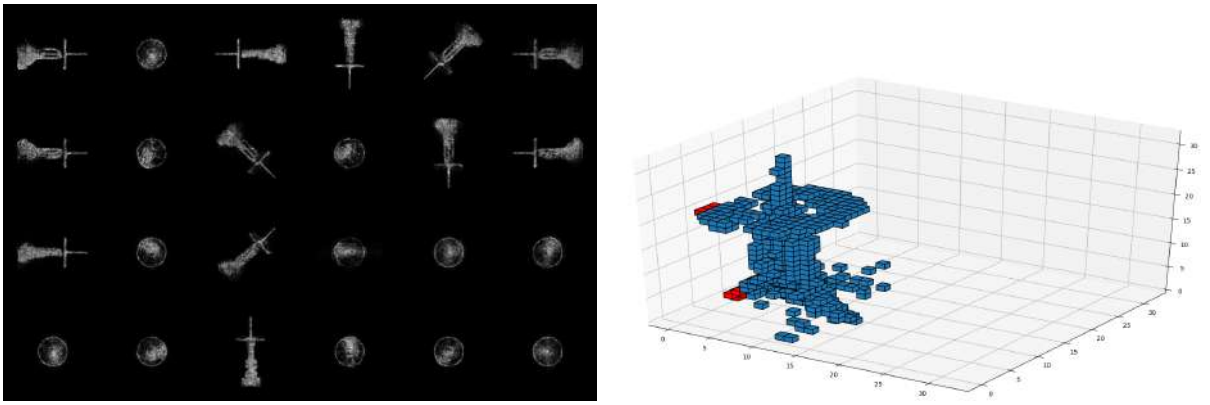


Figure 2.5: Projections of points cloud representing a valve to multiple 2D images (left) and a voxel grid (right).

voxels information to point clouds, SEGCloud [90] interpolates the class of each point depending on their distance to each surrounding voxel. Using a different point of view, VV-Net [66] prefers to encode the local geometry of each voxel using a Variational-Auto-Encoder.

Multi-view methods for classification are more straightforward when projecting the point cloud but loose spatial information in doing so [74][87]. When applied to classification, multiplying the points of view can be done easily and seems to alleviate this problem [87]. However, when applied to semantic segmentation, the back-projection method can be difficult to implement. Snap-Net [15] successfully applies multi-view methods to large scenes by using a complex system of image layers to save image-to-point-cloud correspondence. However, the efficacy of such method in complex and environments with high object density, such as industrial installations, is not studied. As it depends on camera viewpoint, this method could quickly lose efficiency on complex scenes with a high number of objects and where blind spots would be commonplace. However, when a good camera viewpoint is available and coherent with the task at hand, it can be applied with good performance. This is for example the case of coral reef data segmentation [51], where a top-down viewpoint makes sense.

If multi-view is not easily applicable when working with a large scene, when the point cloud is acquired from a single sensor at its centre, image projection works well. In those cases, a spherical projection of the point cloud is used to create a single large image of the scene [67][100]. In this condition, back-projection is easier and the unknown surrounding the camera placement is solved. This method is often used for self-driving applications and allows working in real-time [36].

2.3.1.2 Point-based methods

Point-based methods work directly on the cloud and are classified following the way they extract features from the cloud [40].

The methods from the first category follow the principle used by PointNet [75] and rely on Multi-Layer-Perceptron (MLP) to extract features from a set of points. This has the advantage of extracting good features that are robust to some transformations but loses local spatial information of the cloud, a problem that is the most studied in this family of methods. PointNet++ [76] aggregates features in a method similar to pyramidal pooling to solve this flaw. PointSIFT [49] encodes local features with a method similar to SIFT and PointWeb [109] uses a local point graph before computing features. Some methods will also use recurrence to spatially combine features. For example, RSNet [47] slices the cloud in several parts where features will be independently extracted. The features of each of the parts are then aggregated in a recurrent network for each geometrical axis. However, this technique does not link together each axis which are considered independently. This problem is absent from 3D-RNN [106] which correctly links the longitudinal and lateral axes. However, the operation is not commutative and only the lateral axis will get the information from the longitudinal axis.

Other methods have moved away from MLP and try to define new convolution operations able to operate on point clouds. These techniques are difficult to compare with one another as their only common characteristic in most cases is defining a convolution operation on point clouds without using MLP. As such, they are named as Convolution-based techniques. Nonetheless, some of these newly defined convolutional layers work really well. This is for example the case of KPConv [92], which defines its kernel as a set of weighted points in a sphere and its convolution operation as a distance operation between data points and its kernel. A similar idea is used by A-CNN [52] which projects points on a ring where it performs convolution.

The last category of point-based methods considers the point cloud as a graph and defines convolution as operations on the edges or summits of this graph. In those graph-based methods, methods differ by the graph construction methodology and how they compute convolution. For example, DGCNN [98] creates the graph dynamically following the position of the points in the feature space and applies the convolution along the edge of the graph. This allows the method to link points of similar semantic value in the deeper part of the network. Differently, GACNet [93] works on a more static graph and applies the convolution on the summit. It also uses attention layers to better communicate the relationship between summits. These methods are still considered as point-based as the graph is only a representation of the point cloud. Projection methods can also use graph, such as LatticeNet [8]. However, in the case of LatticeNet, the points are embedded in a lattice graph and not describing an implicit graph used to apply convolution.

2.3.2 General Semantic Segmentation Architecture

The way a deep neural network is constructed, its architecture, defines its properties. Since 2012 and the design of numerous architectures, the notion of deep has also considerably evolved: VGG-16 [84], one of the first architectures, was considered really deep in 2015 with its 13 convolutional layers. In 2017, Inception-ResNet-v2 [88] was composed of more than 130 convolutional layers. These were changes developed in network architecture in this two-year span, such as ResNet [60][88], which enabled the construction of such a deep neural network.

Semantic segmentation is a different task compared to classification, which also induced change in network architecture. The first fully convolutional network for semantic segmentation [81] tried to "combine what and where" by grafting upsampled prediction made by convolutional layers to the convolutional part of a classification network. This showed that a change in architecture was necessary for semantic segmentation network but also the advantages with working on fully convolutional network (absence of constraints on input size). Around the same time, the U-Net architecture [79] was proposed, which is composed of an encoder and a decoder part. The encoder part looks like the convolutional part of a classification network and focuses on extracting features from the input. The encoder uses pooling layers, which reduce the input size. The decoder part then takes the extracted features and upsamples them until they are back to the input original size. A prediction can then be made on this map of features. This architecture is the one most commonly used in semantic segmentation nowadays. U-Net [79] prevents a loss of information between the encoder and decoder part by linking each encoder layer with its decoder counterpart and giving via this link the data before pooling to the decoder. A more modern adaptation of this architecture is SegNet [11] which improves the upsampling technique.

Other improvements were also made on semantic segmentation, such as using atrous convolutions to increase the reception fields of convolutional layer [26]. This shows the need for segmentation networks to have a large spatial awareness. An increase in kernel size can achieve a similar effect [72]. Increased spatial awareness can also be attained by using Spatial Pyramid Pooling [45] which replaces the common U-Net architecture in a work made by Zhao et al. [110].

2.3.3 Point Cloud Semantic Segmentation

Now that the required concepts of applying deep learning to point clouds and the architecture of a neural network doing semantic segmentation were presented, it is possible to go further into the subject of Point Cloud Semantic Segmentation (PCSS). The datasets used in the literature are first presented in Section 2.3.3.1 before a comparison of state-of-the-art PCSS methods is done in Section 2.3.3.2.

2.3.3.1 Datasets

Deep learning techniques need a large quantity of data to work properly, as such, having highly informative datasets is necessary to develop new techniques. In the literature, two kinds of datasets are mostly used. The first ones represent interior scenes, the second urban environments. These two show the domains on which most research on PCSS is done: robotics and autonomous driving.

Before the apparition of the current datasets, which contain numerous labelled points, most 3D datasets were either composed of RGB-D images [83][85] or few points [78][63]. Each dataset presented hereafter contains more than one hundred million annotated points.

INTERIOR DATASETS For interior scenes, two datasets are mostly used: S3DIS [10] and ScanNet [30].

S3DIS First introduced by Armeni et al. in 2016, S3DIS[10] is one of the most commonly used datasets in the literature. This dataset is divided into 6 different zones representing an office-like space and contains 273M points. Composed of 13 classes, this dataset is particularly imbalanced with simple classes (floor, wall, ceiling) being the most common.

Generally, two kinds of semantic segmentation tests are carried out on this dataset:

- 6-fold test, where the method is trained on a random sample of rooms of each 6 areas and tested on the remaining rooms.

- "Area 5" test, where the method is trained on areas 1, 2, 3, 4, 6 and tested on area 5 which is the largest and present a great variability on each room layout.

Despite its use in the literature, this dataset presents three drawbacks. First, as it is acquired from a single location, objects of the same category are quite similar to each other. Secondly, some categories are much harder to define and detect such as the beam and column classes that are indistinguishable from their surroundings at first glance and scarcely present in the dataset. Lastly, the dataset was acquired with the Matterport Pro 2 RGB-D camera, which output meshes. These meshes are then sampled to produce the point cloud used for semantic segmentation [9]. The acquisition approach is problematic as it only produces some of the commonly found acquisition artefacts present in point clouds. Occlusions are present but the noise on point positions is reduced compared to photogrammetry and lasergrammetry approaches. Moreover, point distribution (position and density) is different from both direct acquisition approaches.

ScanNet Acquired with RGB-D sensors, ScanNet[30] was conceived more as a successor of Nyu V2[83] and Sun RGB-D[85] than a full point cloud dataset. However, it is still used considerably as a PCSS dataset as it contains more classes than S3DIS. Its popularity is mainly due to its size (2.5M RGB-D images) and the data available: camera poses, 3D surface reconstruction, textured meshes and aligned CAD models.

Due to its data being RGB-D, some methods transform the data back to voxels to evaluate their performance while others only consider the points.

URBAN DATASETS Three main datasets describe urban environments. A first dataset, Semantic3D [41]s focuses on the general comprehension of the urban environment. The other two datasets, SemanticKITTI [14] and NuScenes [17], focus more on autonomous driving

Semantic3D Semantic3D was the first large, publicly available, point cloud dataset directly acquired [41]. Previous datasets were either not made available, with a low point density or few points. It contains 8 semantic classes and roughly 4 billion points divided in 30 scenes. Each of its scenes is acquired with a static laser scanner. The classification used focuses more on the composition of an urban environment than autonomous driving. For example, only one class considers road objects (cars) whereas the distinction between man-made and natural ground is done. An interesting challenge of this dataset is its high number of occlusions appearing due to the acquisition method employed.

This dataset was used a lot after being published but is less used by the most recent works. In the literature, two tests can be carried out on this dataset: the full test, where 15 scenes are used and the reduced test where only 4 scenes are used.

SemanticKITTI The current reference dataset for urban LiDAR dataset, SemanticKITTI [14][13] contains a number of points comparable to Semantic3D [41]. However, SemanticKITTI offers a holistic dataset with labels available for semantic and panoptic segmentation but also semantic scene completion or moving object segmentation. The number of classes is also more important: 22 classes divided in 7 broader categories.

This dataset is based on odometry data [37], as such a high number of scans is available, most of them overlapping with each other. In total, 22 sequences are available, representing each vast scenes.

NuScenes NuScenes [17] also focuses on urban environment, particularly on autonomous driving. This dataset presents itself as the successor of SemanticKITTI [14] and contains both more points and more classes than the previous dataset. However, as NuScenes is relatively recent compared to SemanticKITTI, it is less widely adopted.

2.3.3.2 PCSS Methods

In this part, a comparison between the existing PCSS methods is presented using the datasets presented in Section 2.3.3.1 and the metrics defined in Section 2.2.2. Most methods follow the common semantic segmentation network architecture detailed in Section 2.3.2 and follow the classification of Guo et al. [40] presented in Section 2.3.1.

An overview of the methods available in the literature since 2017 is presented in Table 2.1. Methods difficult to classify are attached to the broader categories of point or projection based. This is the case for HybridCR [57] which defines a framework enhancing point-based network performances and LatticeNet [8] whose method of projection onto lattice is unique in the literature.

As could be predicted from the weakness of each type of feature extraction methods, most PCSS methods are point-based. Upon the apparition of PointNet [75] and subsequent point-based methods, research interest appeared to diverge from projection-based methods for PCSS. They are mostly found in applications where the point cloud is centred around the sensor and time constraint is important, such as in autonomous driving. For example, RangeNet++ [67] and SqueezeSeg [100] take advantages of such constraint to project the point cloud onto a spherical image. A similar situation also occurs in detection methods, a task of similar complexity as semantic segmentation [36].

For point-based methods, MLP are prevalent. We feel that it is due to the historical importance of PointNet [75] and PointNet++ [76] as well as the simplicity in using MLP. Seeing their performance, this seems to balance the disadvantage in losing the benefits of purely convolutional segmentation networks.

Looking at the latest works in the literature (Tab. 2.1), most seem to focus either on sensor fusion with projection-point hybrid methods, or on creating real-time semantic segmentation networks. As such, the general increase in performance of PCSS methods slowed down since 2019.

2.3.3.3 PCSS Input Format

In the literature, when methods are described, the point of focus is mostly the kernel used as well as the general architecture of the network. As such, few works innovate in terms of data processing. When looking at all point-based methods presented in Section 2.3.3.2, most use a process similar to PointNet [75] (type of data used, a pillar-based division of scene with a limited number of point per chunk). An overview is presented in Figure 2.6 with detailed tables available in Appendix B (Tab. B.1 and Tab. B.2).

A large part of the literature uses geometrical data (XYZ) and, when available, colour data (RGB) to train their network. When the S3DIS dataset [10] is concerned, the point coordinates normalised by room are also used. These consist of three values between 0 and 1 describing how far the point is in the room compared to an origin point. The same

Paper	Date	Type	S3DIS Area 5		S3DIS 6-fold		ScanNet			Semantic 3D full		Semantic 3D reduced		SemanticKITTI
			OA	mIoU	OA	mIoU	OA_{point}	OA_{voxel}	mIoU	OA	mIoU	OA	mIoU	mIoU
PointNet [75]	2017	MLP	75.0	43.5	78.6	47.7	73.9	52.6	-	-	-	-	-	-
PointNet++ [76]	2017	MLP	-	-	75.7	-	74.3	84.5	38.3	-	-	-	-	-
[34]	2017	MLP	-	-	81.1	49.7	63.4	-	-	-	-	-	-	-
SEGCloud [90]	2017	Voxel	-	48.9	-	-	-	-	-	-	-	-	61.3	-
SnapNet [15]	2018	Multi-View	-	-	-	-	-	-	-	91.0	67.4	88.6	59.1	-
LSP [55]	2018	Graph	86.4	58.0	85.5	62.1	-	-	-	92.9	76.2	94.0	73.2	-
VV-Net [66]	2018	Voxel	-	-	87.8	78.2	-	-	-	-	-	-	-	-
3D-RNN [106]	2018	MLP	85.7	53.4	86.9	56.3	76.5	-	-	-	-	-	-	-
PointSIFT [49]	2018	MLP	-	-	88.7	70.2	-	86.2	41.5	-	-	-	-	-
SGPN [95]	2018	MLP	-	-	80.8	50.4	-	-	-	-	-	-	-	-
PointCNN [58]	2018	MLP	-	-	-	65.4	-	85.1	-	-	-	-	-	-
RSNet [47]	2018	MLP	-	51.9	-	56.5	-	-	39.4	-	-	-	-	-
JSIS3D [73]	2019	MLP	-	-	87.4	-	-	-	-	-	-	-	-	-
ASIS [96]	2019	MLP	86.9	53.4	86.2	59.0	-	-	-	-	-	-	-	-
PointWeb [109]	2019	MLP	87.0	60.3	87.3	66.7	-	85.9	-	-	-	-	-	-
DGCNN [98]	2019	Graph	-	-	84.1	56.1	-	-	-	-	-	-	-	-
GACNet [93]	2019	Graph	87.8	62.9	-	-	-	-	-	-	-	91.9	70.8	-
PointCONV [102]	2019	Conv	-	-	-	-	-	-	55.6	-	-	-	-	-
A-CNN [52]	2019	Conv	-	-	87.3	69.2	-	-	-	-	-	-	-	-
KPConv [92]	2019	Conv	-	67.1	-	-	-	-	68.6	-	-	-	74.6	58.8
MV-PointNet [48]	2019	Hybrid	88.1	62.4	-	-	-	-	64.1	-	-	-	-	-
UPBF [28]	2019	Hybrid	-	-	-	-	-	-	63.4	-	-	-	-	-
SAN [18]	2019	MLP	-	-	78.4	-	-	85.1	-	-	-	-	-	-
RandLA-Net [46]	2020	MLP	-	-	-	-	-	-	-	-	-	94.8	77.4	53.9
LatticeNet [8]	2020	Projection	-	-	-	-	-	-	64.0	-	-	-	-	52.9
$(AF)^2$-S3NET [27]	2021	Hybrid	-	-	-	-	-	-	-	-	-	-	-	69.7
MSSCN [33]	2021	MLP	-	-	89.8	-	-	86.3	-	-	-	-	-	-
FPVC [97]	2021	Hybrid	-	61.7	-	-	-	-	-	-	-	-	-	
HybridCR [57]	2022	Point	-	65.8	-	70.7	-	-	59.9	-	-	-	77.7	54.0

Table 2.1: Overview of the Point Cloud Semantic Segmentation methods available in the literature.

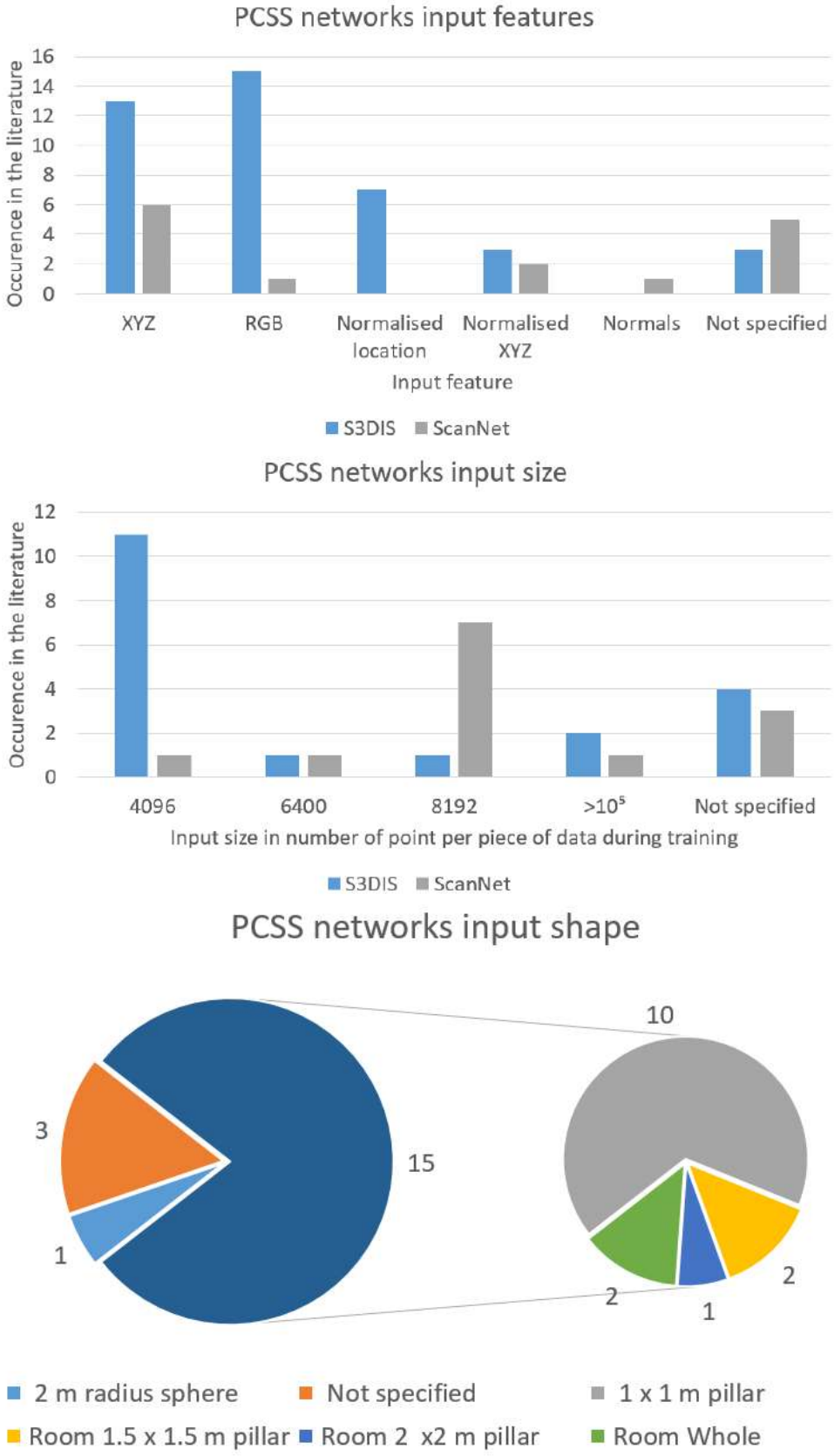


Figure 2.6: Distribution of common input features, size and shape for PCSS networks in the literature. In the last figure, only the S3DIS is taken into account. Twenty and twelve methods were used for S3DIS and ScanNet respectively.

process of normalising geometrical data is often used even when not working in S3DIS. This normalisation can be computed when the whole scene is considered but also on each data chunk. This practice of normalising point coordinates questions the ability of these methods to cope with point translation. And, as seen by Lin et al. [61], some methods are indeed not robust to translation whereas others integrate such robustness directly in their kernel, such as KPConv [92]. Lastly, only one method, A-CNN [52], declares using point normals as a feature in the input of their network.

Most methods only consider a limited number of points at a time during training. This is justified by the cost of most 3D operations applied to point clouds. Computing proximity between points or their neighbourhood is always costly. As time went on, most methods use more points during training, either thanks to the increase in hardware computational power or by devising new techniques. An outlier is clearly RandLA-Net [46] which can process 10 times more points at once than other methods. This is done by dropping precise but computationally expensive methods in favour of more lightweight ones. For example, Farthest Point Sampling is switched with a random sampling.

This limited number of input points also implies the necessity to divide each scene in different part. This is prevalent in methods which only use few points. For S3DIS [10], methods which uses more than 10^5 points (or do not specify) are often training their network on the whole room. Which is, in the case of S3DIS, the most common type of scene used. Except for one case, KPConv [92], which divides the scene using spheres, the others divide the room using square-based vertical pillars of various size. Most of the time, the exact method used to create those data chunks is not specified. It is thus difficult to determine if the division is a pre-processing process or made dynamically during training. Concerning the other datasets, the division process is roughly the same. ScanNet [30] uses slightly larger pillar (1.5×1.5 m pillars in most cases). In the few cases where information is given for SemanticKITTI [14] and Semantic3D [41], either a whole scan or larger pillars (4×4 meters) are considered.

Finally, only four methods mentions their use of data augmentation. PointCNN [58], KPConv [92] and MV-PointNet [48] apply a rotation to the data chunk before feeding it to the network during training. The last case, A-CNN [52], permutes points and scales the pillars to a height of 2 meters. This lack of use (or information) is quite concerning as data augmentation is more widely used in 2D images and proved its worth.

2.3.4 Synthetic data augmentation

Data are at the centre of every machine learning approach, even more so when considering deep learning that heavily depends on their abundance. As seen in Section 2.1.2, there is a severe lack of PCSS datasets representing industrial installations. One way to alleviate this problem is to use synthetic data in addition to the available acquired data.

One problem with synthetic data is their difference with real data, which makes it harder for the network to learn. This difference between real and synthetic data, named "domain-shift", makes it harder for the network to generalise on real data [80]. Most techniques whose goal is to reduce this domain shift focus on modifying the neural network [80]. However, few techniques look into reducing the domain shift by working directly on the data, more so in the case of point cloud data [103].

2.3.4.1 Synthetic data generation

The lack of research in this domain seems to be due to the current 3D synthetic data creation techniques. Few works focus on point clouds and most create 3D synthetic data for 2D or RGB-D applications. As such, we will include such methods in our analysis.

Two philosophies of 3D synthetic data creation techniques exist:

- Using already existing 3D data present in other domains[100][101][103]. For example, SqueezeSeg [100] uses a video game to create synthetic data for autonomous-vehicles application.
- Generating scenes randomly by using a 3D objects library [65][59][31].

These methods still engender an important domain shift as a result of their design. Using video games allows accessing quality content but can be limited in the variability of the model available or the API (Application Programming Interface) of the game. In the case of SqueezeSeg [100], bicycles could not be generated from the game due to possessing a cylindrical hit box: their ray-tracing method fails to acquire the corresponding 3D model. Moreover, their ray tracing method does not include a noise model, which is added in post-processing. This further increases the domain-shift. The second version of this method proposes to add laser intensity information via a learned model but does not look into reducing domain-shift before training [101].

When considering the use of CAD data to create synthetic data, only Noichl et al. [69] seem to look into this topic. However, they concluded their research into studying the quality of the point cloud created by sampling CAD models with a virtual laser but did not extend their research to semantic segmentation.

One work tries to create methods to bridge the gap between acquired and real LiDAR data[103]. This point cloud translation method [103] is divided into two generative models (GAN). One focuses on modifying the general appearance of a synthetic point cloud to look alike a real one. This part is used both as a noise model and a correction on the geometry of the point cloud. The second part decimates the synthetic point cloud to achieve a sparsity of points similar to acquired data. As sparsity is a difficult concept to grasp in point clouds, this GAN is trained on projected images. This sparsity model is able to recreate some peculiarities of real-world objects. For example, glasses of cars

that are not correctly processed by LiDAR. If this method significantly bridges the gap of synthetic points geometry, it is tailored for navigation application where the sensor is at the center of the scene.

2.3.4.2 Colouring point cloud

When synthetic data are created, colours can sometimes be made available, for example, when a video game is used or textured models are associated to mesh models in a 3D library. In the case of CAD models, this is often not the case. As such, the generated synthetic data is colourless.

With RGB-D data, DE²CO [21] shows that it is possible to transfer information via colouring. In their case, depth information is transferred to the colour channels. This information transfer capacity is interesting, as it does not need semantically labelled data. However, few works exist in point cloud colouring. They all focus on creating a realistic colouring of point cloud and use Generative Adversarial Networks (GAN). Most are only able to colour single objects [62][19] and fail to correctly process whole scenes [82]. If Point2Color [82] possesses such capacities, it needs to combine corresponding images information with the point cloud to work. As such, it is out of scope in our case. Finally, style transfer networks working on point cloud still need a base colour information [20] and are not yet able to only use geometrical information.

2.3.5 Discussion

Current point cloud semantic segmentation methods are effective enough when one needs to work on a case similar to those present in the literature. However, the available datasets are scarce and the domains of application represented are even scarcer. Moreover, most works are focused on solving the task of applying deep learning to a point cloud and thus on creating new feature extraction methods, point cloud processing kernels or network architectures. Works focusing exclusively on data are almost absent from the literature. Furthermore, the influence of the shape of point clouds given to segmentation networks is not studied.

Synthetic point clouds can be used to alleviate a lack of data. However, the methods to generate such point clouds in the literature all use existing mesh datasets. Using a 3D scene generation algorithm made to create other types of synthetic data, such as 2D or RGB-D images, presents other challenges, notably realism of the obtained scene. Finally, few methods focus on the mesh-to-point-cloud transformation.

2.4 Conclusion

In this part, the current application of computer vision techniques to the industrial sector was first studied. The recent publication of the ISO-19650 norm corroborates the motivations behind the project containing this thesis. However, when looking at the scientific literature, few works on PCSS applied to the industrial domain are published. Complete, available and large PCSS datasets representing industrial facilities are also absent. Nonetheless, this allowed us to confirm the utility of applying PCSS to the industrial sector and offered insight on which classes of object segmentation efforts are focused.

After defining the problem of point cloud semantic segmentation more formally, a study of PCSS was conducted. From this, a categorisation of current state-of-the-art methodologies was deduced. Applying deep learning to point cloud is a partially solved problem. Current methods work well on the most used datasets. However, these same datasets only represent two environments of application: interior and urban. They also only contain a partial representation of the real world as they are acquired from only one or two different locations. When the environment of application becomes less controlled, there are few guarantees on these methods performances. Furthermore, if the scope of research recently extended from applying deep learning on point clouds to more topics such as real-time applications or sensor-fusion, some areas are still set aside. This is the case for the pre-processing of point cloud data, for which few studies exist.

Lastly, a study on the topic of synthetic data augmentation was made. If evidence points towards the ability of this method to alleviate our lack-of-data problem, the method of application to our domain is an open problem. Most synthetic data generation processes focus on creating 2D or RGB-D images and the transformation process from mesh to point cloud is poorly studied, notably its influence towards PCSS performance. One existing work [103] hints at the importance of this process even if the method used is not applicable in our case. Finally, a potential way to improve synthetic data, colouring, was studied. Similarly, if a work on RGB-D images [21] hints at the interest of such process, the research on point cloud colouring is still in its infancy and no applicable solution exists yet.

Chapter 3

Working with data acquired from an industrial setting

This chapter lays the foundation for the work made in this thesis. It introduces a classification of industrial objects and determines which categories of objects are to prioritise for semantic segmentation. The peculiarities of data acquired from an industrial environment are then shown: the high variability in object relative size but also the scene layout, which differs greatly from the common applications of PCSS in the literature. Drawings from these points, the general method explored in this thesis is presented. It consists of a data processing framework divided in two parts. The first is specific to synthetic data and focuses on increasing their quality. The second part is applied to both acquired and synthetic data and tries to increase data informativeness by applying different pre-processing techniques.

By taking into account the current state-of-the-art and the global project goal, objects of interest will be first identified in Section 3.1. The challenges associated with working with industrial data are presented in Section 3.2. To conclude this chapter, an overview of the proposed solution and methodology to tackle these challenges will be presented in Section 3.3. Common settings between most experiments are then reported in Section 3.4, such as the segmentation network and the metrics used.

3.1 Industrial objects of interest

In this part, objects of interest for the semantic segmentation task, as defined in the objectives in Section 1.3, will be detailed. Two categories of objects are considered as a priority for segmentation: piping and structural objects, as they represent the bulk of objects in an industrial scene. Other categories of interest include industrial equipment, the electrical network, instrumentation and general architectural objects such as walls and

Object type category	Examples
Structural	T-brace, angle, wide flange, circular hollow, handrails, T shape, I-beam, block, plate
Equipment	Vessel, exchanger, pump, compressor, tank
Piping system	Tee, elbow, valve, flange, reducer, pipe
Electrical	Cable tray, conduit, electrical panel, lights
Safety	Fire suppression piping, fire extinguisher, signage
Instrumentation	Barometer, sensor, controller
Heating, Ventilation and Air Cooling (HVAC)	Pipe, duct, elbow, flange, valve
Architectural	Wall panel, slab, window
Civil	Foundation, bollard, barrier, kerb

Table 3.1: General industrial object classification following [6]

ceilings. This categorisation of objects of interest can be extended to specific equipment and piping systems, such as in the work of [6], displayed in Table 3.1. A illustration of most of these objects is provided in Appendix C.

The objective of the overall project SMARI being a gain in modelling time of existing industrial facilities, it is necessary to reconstruct most objects reliably. The status of priority targets of piping and structural elements is due to their general geometric simplicity and recurrence in the industrial scene. Other categories are more difficult to segment, as they are either underrepresented or more diverse in shapes. Architectural elements may be among the easiest of the remaining categories as they are still represented by numerous instances and are simple objects in most cases, such as concrete slabs. Pieces of equipment can be quite numerous in an industrial setting but are too diverse to be reliably modelled. An exception could be found in large devices such as tanks and vessels, whose large size and low complexity can represent a good trade-off when considering the segmentation step. Finally, most objects of the instrumentation and electrical categories are highly under-represented to be considered a priority. This problem of representation comes from both the lack of instances and the small size of objects in this category that impedes acquisition.

However, a reliable reconstruction does not only come from correctly modelling and labelling objects. A definition of model reliability could be found in the concept of Level Of Development (LOD)[12]. A LOD can be seen as the degree of reflection put in a model: how well it is integrated in its general project, how each object takes into account its surroundings and how definitive elements of the model are. A summary of LOD levels is given in Table 3.2.

Considering these definitions, the goal of the overall SMARI project is to obtain a LOD 350 model of the acquired industrial scene. The acquired scenes are already constructed,

Level	Official definition	Interpretation
LOD 100	The Model Element may be graphically represented in the Model with a symbol or other generic representation, but does not satisfy the requirements for LOD 200. Information related to the Model Element (i.e. cost per square foot, tonnage of HVAC, etc.) can be derived from other Model Elements.	LOD 100 model elements are purely informative and any information attached is approximate. At this level, only the existence of the element can be considered as reliable information.
LOD 200	The Model Element is graphically represented within the Model as a generic system, object, or assembly with approximate quantities, size, shape, location, and orientation. Non-graphic information may also be attached to the Model Element.	LOD 200 model elements still contains approximate information but are graphically represented. This representation can range from a recognisable placeholder to a block used to reserve space for the element.
LOD 300	The Model Element is graphically represented within the Model as a specific system, object or assembly in terms of quantity, size, shape, location, and orientation. Non-graphic information may also be attached to the Model Element.	LOD 300 model contains precise information about general shape, position and orientation but without consideration for surrounding elements or composition. Information about the object is reliable when connections to other components or details of the object components are not of importance.
LOD 350	The Model Element is graphically represented within the Model as a specific system, object, or assembly in terms of quantity, size, shape, location, orientation, and interfaces with other building systems. Non-graphic information may also be attached to the Model Element.	LOD 350 model elements augment LOD 300 models by adding consideration to internal composition and connection to the surrounding environment.
LOD 400	The Model Element is graphically represented within the Model as a specific system, object or assembly in terms of size, shape, location, quantity, and orientation with detailing, fabrication, assembly, and installation information. Non-graphic information may also be attached to the Model Element	LOD 400 model elements contains sufficiently reliable and detailed information to be ready for fabrication.

Table 3.2: LOD definitions following [12]

as such, they should contain sufficient information to go beyond LOD 300. Creating LOD 400 models is the task of industrial draughtsmen and should be kept as such. LOD 400 models contain pieces of information which go much farther than merely storing the facility description, such as fabrication and assembly.

To achieve LOD 350 with the overall solution, the segmentation step must be able to process the whole information of the acquired point cloud and determine which piece of information is sufficiently important to keep and which information to discard. When considering the piping and structural objects categories only, it is clear that other categories must be taken into account to achieve the desired level of precision. Architectural objects and equipment must be taken into account to model piping and structural elements interface with other building systems. Ideally, interface with instrumentation and electrical objects should be modelled and will thus be considered for semantic segmentation. However, lack of data quantity and reliability will prove to greatly impede the segmentation of these classes.

The observations made in this section are taken into account when creating the SMARI semantic segmentation dataset, described in Section 3.3.1.

3.2 Particularities and constraints of the industrial setting

As seen in Section 2.3.3.1, the settings used in the literature are quite different from an industrial environment. This change in setting does not only change the classes which need to be segmented but introduces new challenges to be tackled in order to perform a correct semantic segmentation. This section focuses on presenting these challenges which were summarised in [22]. When excluding the difficulties in creating a new dataset, two constraints need to be considered: the peculiar layout of industrial settings (Sec. 3.2.1) and the significant variability in relative object size (Sec. 3.2.2).

3.2.1 Industrial scene layout

There are three major challenges arising from an industrial setting layout. They are the consequences of not being sensor-centred, having a scene structure which varies greatly depending on its purpose and containing zones of high complexity.

When a scene is acquired, there are two possible sensor placement strategies, depending on the goal behind the acquisition.

First, only one sensor is used, the scene is centred around this sensor and some occlusions are acceptable. This is the paradigm used when the goal is navigating within the scene. As the sensor is present within the object, only its immediate surroundings are considered. Moreover, thanks to the mobility of the sensor, multiple sensor-centred

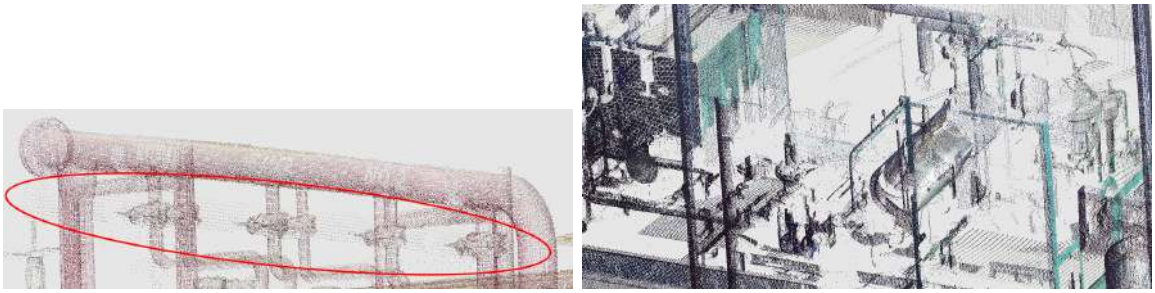


Figure 3.1: Illustrations of complications due to the data acquisition process, on the left a "phantom" object, on the right large-scale occlusions.

viewpoints can be considered (such as with a multi-view hypothesis fusion [70]). Datasets [14] [17] representing urban environments are an example of this paradigm, their intended goal being autonomous vehicle navigation.

In the second case, multiple sensors and/or viewpoints are considered at the same time. The goal is to understand the scene. Therefore, the point cloud scene is static and the relation between each sensor position and time of acquisition is unknown. This is our case with an industrial setting but also some interior environment datasets such as S3DIS [9] and ScanNet [30]. Such data is prone to additional risks compared to the acquisition for navigation paradigm. Scanning an object from several points of view can create an overlap noise. This effect can range from small noise indistinguishable from Gaussian noise to the creation of a full "phantom" object shifted from ground truth position. A second risk is the presence of occlusions which can not be recovered from as there is no mobility in the scan process. The point cloud is done as is to the segmentation method, it already happened, and no new information can be expected. When the goal is scene understanding, some occlusions cases can seriously impede the segmentation process. Both those cases are illustrated Fig. 3.1.

A second challenge associated with working with an industrial setting is the variability of possible scenes. Even when restraining the scope to oil & gas, two major variations can occur. One kind of scene is storage areas where mostly piping and large tanks are present. The second kind consists of processing areas where dense clusters of objects are observable. Extreme examples are presented Fig. 3.2. These differences in scene structure will create an imbalance in data as storage areas do not contain every kind of class. A network ability to understand context could be crippled if not enough examples of each scene kind are provided during training. Another common hurdle to these settings is their verticality. If in urban environments most objects of interest are connected to the floor and most interior scenes are only a floor high, objects of interest can be several meters high in industrial settings. As a result, the network attention will be spread over a larger area than what is commonly presented in the literature.

Storage areas are simple enough and do not cause problems during segmentation (excluding big tanks which can look like walls). Processing units add other domain specific

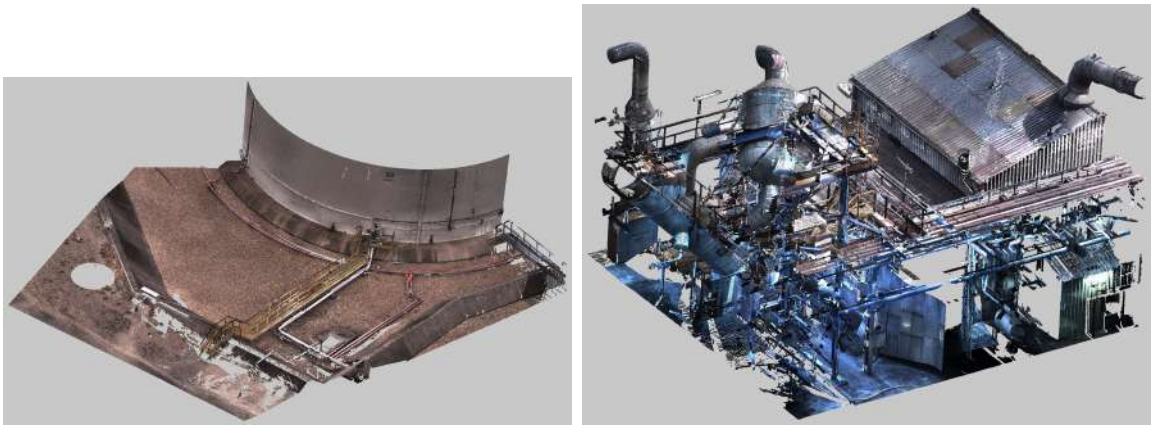


Figure 3.2: Two kinds of scenes in an industrial setting with a high difference in data distribution: a storage area and a processing area.

challenges. They are mostly composed of a dense cluster of objects, some being concave, such as piping. This increases the risk of occlusion during acquisition and make segmentation difficult due to the objects being close to each other. The limit between two objects, such as an electrical cable and a beam, is difficult to find, even via manual segmentation. Furthermore, this density of objects is often translated by a high quantity of objects layer along the vertical axis which further aggravates this specific industrial setting hurdle.

3.2.2 Variability in relative object size

The relative size of objects in an industrial setting is an additional hurdle which impedes segmentation performance. On urban or interior environments, objects of interest are often in the same general size level. More specifically, there is no more than one order of magnitude between the size of two objects of interest. The general size distribution in an industrial environment is more spread out. Examples with tiny objects of interest near much bigger ones exists. As segmentation networks offer a limited resolution in their middle layers. Smaller objects have a risk of being ignored in those parts. Examples of such cases include barometers next to tanks and electric cable on walls. Another problem being that objects of similar categories are also often similarly sized, which creates another risk of data imbalance in industrial dataset due to the acquisition process.

When looking at the volume of each object in our industrial dataset (SMARI), described later in Section 3.3.1, and the S3DIS [9] dataset, the difference is quite clear (Fig. 3.3). When calculating the volume taken by the bounding box of each object, the industrial dataset contains seven times more tiny objects (of volume inferior to 0.01 m^3) than S3DIS. The objects are also more concentrated on the lower side of the spectrum in the industrial case. This confirms the overwhelming presence of small objects. When looking at large ones (of size superior to 1 m^3), only 1.65% of the objects take a volume superior to 5 m^3 in the S3DIS dataset. On the other hand, 2.79% and 4.83% of objects

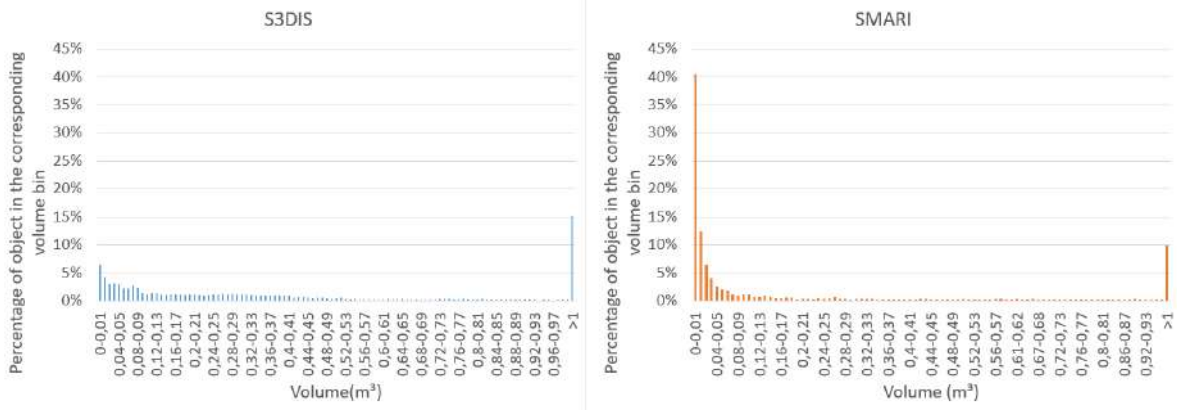


Figure 3.3: Volume (m³) distribution, per object, in the S3DIS and SMARI datasets; floor class excluded.

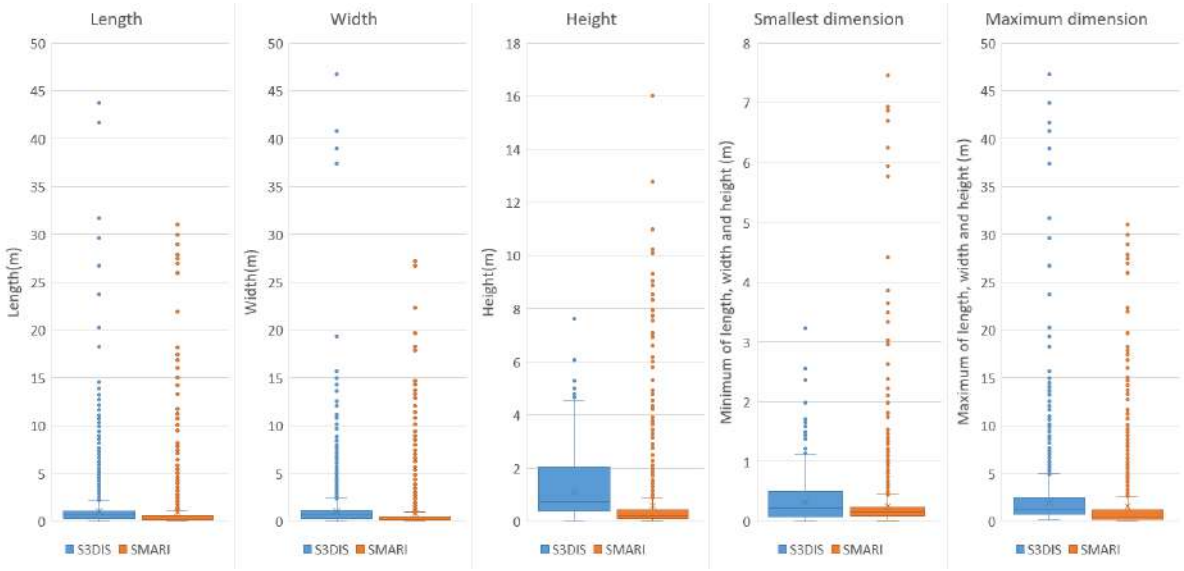


Figure 3.4: Size distribution, per object, in the S3DIS and SMARI datasets; floor class excluded.

are bigger than 5 m³ and 25 m³ respectively in the SMARI dataset. This shows that the volume distribution of industrial data seems to be more spread out and contains more outliers than a typical case of the literature, namely interior scene.

This can be confirmed when looking at the difference in the size distribution of objects Figure 3.4. Most objects in the industrial dataset are smaller, but a greater spread in the minimum size of out of the norm objects exists.

Finally, the difference in distribution between the two datasets (Fig. 3.5), in number of points per class, confirms the increased risk in data imbalance in industrial data.

3.2.3 Discussion

In this section, we underlined the specific challenges of working with data acquired in an industrial setting. This domain generates discrepancies in the datasets, be it in terms of

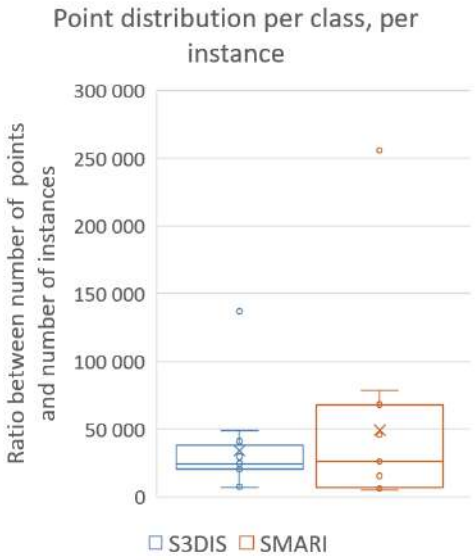


Figure 3.5: Distribution of the ratio between number of points and number of instances per class for the SMARI and S3DIS dataset, floor class excluded. The tank class is excluded for SMARI dataset, its ratio being 1 588 740.

scene structure, point cloud quality, class balance or object size. Moreover, the complexity of each scene will increase the difficulty in manually segmenting data to conceive a proper dataset. All these characteristics call for a more effective approach in data usage.

When looking at popular public datasets, interior scenes seem to be the closest to the industrial setting. They present the same challenges associated to working on a fixed point cloud paradigm. However, this is the only common aspect, as the other peculiarities of industrial scenes are not present in these datasets. They can nonetheless be used as an interesting source of comparison with the literature.

3.3 Proposed method to process available data

On the basis of the previous observations made on industrial data, a general method to process available data in a way that enhances their informativeness is proposed in this section. The data classification used in most of this thesis is presented in Section 3.3.1 before describing an outline of the general data workflow (Sec. 3.3.2).

3.3.1 Data classification

In most works presented in this thesis, a single semantic taxonomy of objects is used. This is presented in Table 3.3 and focuses on industrial draughtsmen needs. Exceptions to its use are in Section 5.3 which focuses explicitly in challenging this classification and during the overall solution validation, Chapter 6, which uses an optimised classification.

This classification is first compared to the works of Agapaki[6][7] in terms of classes

Category	Description	Colour
Pipe	Any piping, with or without protection. Also includes elbow, tee and flange.	
Beam	Major metallic beam supporting the installation.	
Valve	Any type of valves and flange.	
Electrical panel	Any kind of electrical panel and device linked to electrical cable which is not part of another class.	
Instrumentation	Instrumentation mounted on piping, such as a barometer.	
Tank	Tank and liquid storage unit of every size.	
Electrical cables	Electric cable and cable tray.	
Floor	The floor of the scene, natural or artificial.	
Walkway	The complete walkway, with the metallic duckboard and handrail. Also includes vertical stair.	
Miscellaneous	Object that could not be included in other categories. Their utility is let to be determined by the operator of the final product.	
Artefact	Scanning artefact and points which could not be classified	
Piping rack	A rack of piping for which individual pipes could not be seen during the acquisition phase due to occlusions	
Pump	Pump and compressor	
Structural	Structural element excluding beam. Architectural element such as wall, foundation and ceiling.	

Table 3.3: Semantic classification of objects used in this document. The colour given by class is used when presenting segmentation results.

used and number of instances of each class in the SMARI dataset (detailed Table 3.5) before looking into the distribution of points per class.

The proposed classification diverges from the general object type categories proposed by [6] (Tab. 3.1). In our case, objects were not classified based on their general types but following practical concerns of industrial draughtsmen: object recurrence, criticality and cost of modelling. Most recurrent object categories[1] include piping (32.1% of instances) and structural elements (40.5% of instances). Valves being critical components of the piping system, they are considered as a class of their own. Differently, the piping rack class is needed to represent dense clusters of small-bore piping, which differs greatly in their utilisation compared to larger bore piping. When looking at structural elements, they represent 40.5% of all object instances. Walkways are excluded from structural elements, as they are a particular class of structural elements, with a specific functionality and a high modelling cost. Another split was made on the structural category by isolating the most recurrent objects (large beams). On the other hand, architectural elements are included in the structural class as they are represented by fewer instances and most of them are linked to structural elements by their functions. This includes for example foundations of small structural hollows supporting pipes. The only kinds of equipment that present

1. Based on the dataset described Table 3.5

a semblance of uniformity in their shape are tanks and pumps (2.1% of objects). Other miscellaneous equipment represent 9.4% of objects. This category includes mostly objects that are unique. Thus, their importance towards the installation is to be decided by the final program users. As determined in Section 3.1, the instrumentation object category is present in the dataset. Electrical objects were split in two following their general shape and function: cables or panels. Finally, the segmentation process must be able to determine the usefulness of each point in the input data. To answer this requirement, two classes exist in the dataset: the floor and artefact classes. If the floor class can give important information to the network during segmentation and on the reconstructed drawing, artefact concerns the points flagged as "ignorable"[2].

When looking at the statistical distribution of objects reported by [6], similar frequency is observed in the structural and piping class. One major point of difference is the electrical components which are far less represented in the SMARI dataset (6.5% of instances against a frequency of 26.90%). The dataset used by [6] being unavailable, we can only make assumptions regarding the reasons behind this difference. Due to lack of information, [6] could only compute the frequency of electrical class objects in a petrochemical plant which seems to mostly contain processing units. In our data, electrical elements are mostly present in those processing units and not in storage areas (9.2% versus 1.2% of instances). This discrepancy in data origin could explain a part of this difference. Another difference would be the scan quality. Most detected electrical cables are present in cable trays that are large structures containing large number of cables. Smaller cables being occluded by the beams they are running against, they are often absent in lower quality scans.

A last difference between the proposed classification and the one finally retained by [6][7] (Tab. 3.4) is the underlying goal. The CLOI dataset considers industrial **shapes** whereas SMARI considers industrial **objects**. Thus, fewer industrial categories are considered in CLOI (mostly structural and piping) but with a higher level of details (4 classes for structural elements and 4 other classes for piping).

Following this classification, the dataset was regularly augmented with new data during preliminary experiments. In total, 5 and 4 scenes were segmented to create the training and testing datasets respectively. Those scenes were sampled from 6 different locations and represent a variety of environments, acquisition conditions and detail levels. As shown in the overview provided in Table 3.5, it contains 2.2 times fewer points than the S3DIS dataset [9] (124M versus 273M points), covers a surface 2.5 times smaller (3 555 versus 8 959 m^2) but presents a much more serious case of data imbalance (Fig. 3.6). Even when taking into account the small imbalance in the S3DIS dataset, good results can still be obtained as only one location was used to create it. This can be attributed to the dataset being quite homogeneous in terms of objects represented, their relative configuration and

2. Determining the way by which those points must be processed is out of the scope of this thesis, the interactions with future users of the global solution are not yet tested.

Object class	Associated type category
Angle	Structural
Channel	Structural
CHS	Structural
Conduit	Electrical
Elbow	Piping
Flange	Piping
I-beams	Structural
Pipes	Piping
Valves	Piping
Other	Others

Table 3.4: Classification used in the CLOI dataset [7]

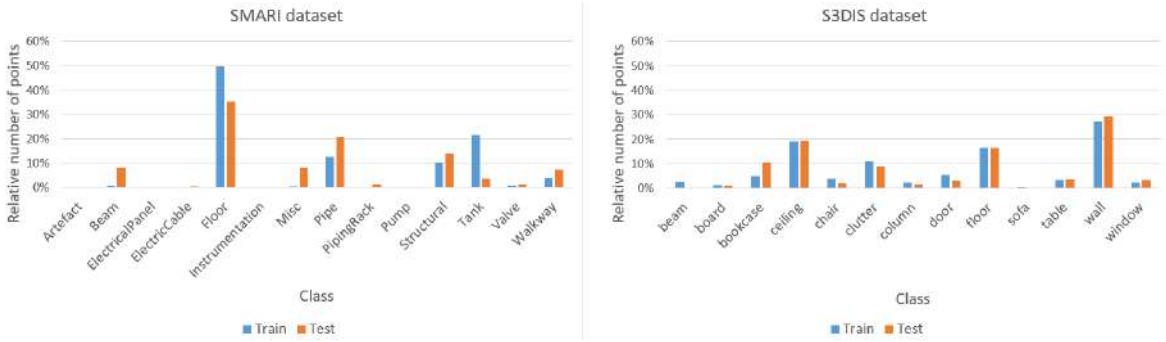


Figure 3.6: Class distribution in the SMARI and S3DIS datasets expressed in relative number of points per class.

colour (this last point is further studied in Section 4.3). The SMARI dataset being heterogeneous in every aspect, a number of point equivalents to the Semantic KITTY dataset, where locations are more diverse, would be needed to obtain similar segmentation results.

Despite its drawbacks, which were mostly felt after several experiments, this classification was used in most tests of this thesis to ensure results consistency.

3.3.2 Data processing workflow overview

By taking into account the specificity of industrial settings seen previously (Sec. 3.2) and the proposed data classification (Sec. 3.3.1), a general workflow to process data can be envisioned. This workflow is described in Figure 3.7 and consists of two parts. The first is applied scene by scene and changes according to each scene source. The second processes the dataset dynamically during training.

The goal of the first part is to take advantage of the several kinds of data available, without limiting oneself to point cloud, in order to create a dataset as diverse and balanced as possible. It is in this part that a difference is made between acquired and synthetic data. Once this step is carried out, no distinction based on data origin should be made

Scene Name	Type	Acquisition technique	Size(m^2)	Number of points
		Train		
Storage 1	A storage area, relatively empty	Lasergrammetry	794	19 410 576
Storage 2	A storage area, with more piping and complexity	Lasergrammetry	779	30 145 820
Unit 1	A processing unit behind a walkway	Photogrammetry	16	3 003 562
Processing 1	A mix of a processing and storage area	Lasergrammetry	1 008	36 793 526
Valves	A close view of some piping	Lasergrammetry	4	10 155 021
Total			2601	99 508 505
		Test		
Piping	An intersection of two piping lines	Lasergrammetry	503	8 622 186
Unit 2	A processing unit	Photogrammetry	40	5 346 637
Storage 3	A simple storage area	Lasergrammetry	279	1 384 835
Plant	A plant with lot of structure which only contains a single colour channel	Lasergrammetry	132	9 352 930
Total			954	24 706 588

Table 3.5: Description of the SMARI dataset, acquired data part

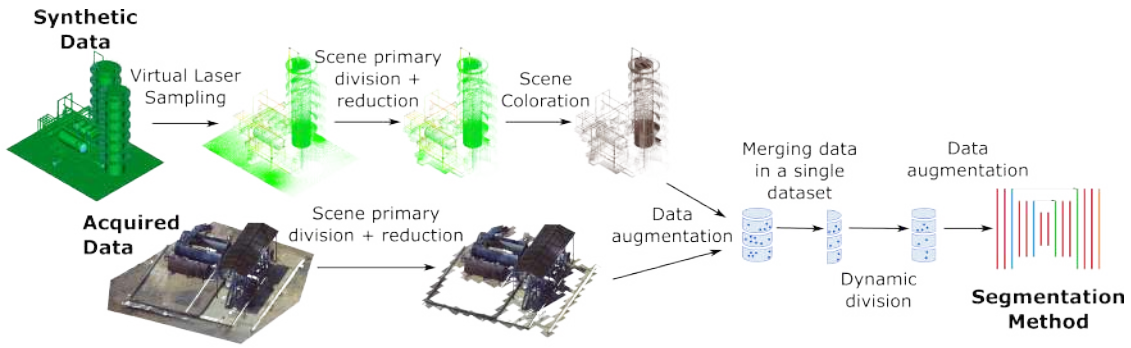


Figure 3.7: General data processing workflow proposed in this thesis.

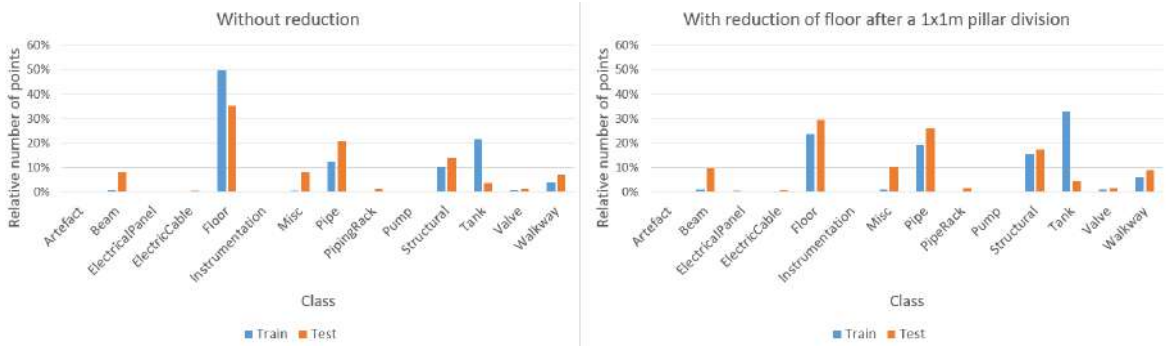


Figure 3.8: Class distribution in the SMARI dataset expressed in points per class, before and after applying a reduction on the floor class.

by the data processing workflow and the segmentation network[3].

For synthetic data, once it is generated and labelled, this part works as follows:

1. The synthetic data in a mesh form is sampled to create a corresponding point cloud.
2. The scene is divided in several parts. The size and shape used depends on the dynamic division parameters of the second part.
3. Parts that do not contain objects of interest following the division are removed from the data. This particular step is called reduction in the following of the document.
4. A colouring model, trained on unlabelled acquired point clouds, is applied to the synthetic data.

The first step is mandatory to operate on a point cloud and offers opportunities to improve the informativeness of synthetic data. The fourth step tries to extract information from acquired point clouds that could not be labelled. Instead of letting them be unused, potentially interesting colour information could be extracted from them. The second and third steps are also considered when working with acquired data. Most scenes are far too large to reasonably fit in the segmentation network[4] input and smaller chunks must thus be considered. Depending on the dynamic division applied in the second part, this division can be pillar, cylinder, cube or sphere shaped. The size of each chunk once divided is also linked to the dynamic division. The reduction is the step used to alleviate class imbalance. Following the scene origin, different classes can be considered as not interesting for the network when they are the only ones present in a chunk. This is the case for the floor class in every case. Synthetic data add the tank class to this list. As illustrated in Figure 3.8, this step alleviates the most blatant case of imbalance at the cost of a reduced dataset (30% of points and 2027 m^2 are removed during this step for the acquired dataset). The beneficial effect of this step on segmentation results was discovered during preliminary tests and is verified on the complete solution in Chapter 6.

3. Though a trained eye can still easily distinguish the two in most cases

4. If a fully convolutional point cloud semantic segmentation network can theoretically work on a whole scene, time, dataset size and hardware capability severely limit input size. Working on division is therefore a study of this trade-off.

The aim of the second part is to increase data informativeness by applying simple but efficient techniques. This enhanced informativeness can be used to increase segmentation performance and segmentation robustness against a Euclidean transformation of the point cloud, such as rotation. This second part is applied right before the input layer of the segmentation network. At first, some data augmentation methods can be used to compute additional information on data[5], such as point normal or curvature. It is followed by further extracting dynamically from each chunk of data a smaller one by a division process. The goal of this step is to increase the variability in the data fed to the network. This step also ensures that a reasonable point density is kept, as it is shown to play a role in segmentation performance (Sec. 5.1). Other data augmentations can then be applied dynamically to this sampled data before feeding it to the network.

The first and second parts of the methodology are detailed and tested in Chapter 4 and in Chapter 5 respectively.

3.4 General experimental methodology

To test the proposed data processing workflow and the hypotheses considered in its creation, several experiments were carried out. This section describes the general conditions of these experiments. This includes description of hardware and software in Section 3.4.1, of the segmentation network used in the experiments (Sec. 3.4.2) and specific metrics considered when analysing the results obtained (Sec. 3.4.3).

3.4.1 Hardware and software

Three different hardware configurations were available during the thesis and are described in Table 3.6. The three run on Windows 10 operating system. The development computer was used for some preliminary experiments and the first computation machine was available since the first experiments presented in this document. The last machine was made available in November 2022. No distinguishable differences in computation results were observed between the three.

Regarding software use, three different languages were used during methods implementation: Python, C# and C++. Python was used for data pre-processing, network definition and training as well as results collection and analysis. C# was used for pre-processing and result collection and organisation. C++ was used in some synthetic data generation process. A detailed list of languages, libraries and software used is presented

5. This concept of static data augmentation move slightly away from the most common concept of data augmentation in the literature, which is dynamic in nature. However, lots of additional information can be added to point clouds before training, for example those proposed by Hackel [42], but they are too computationally expensive to do live during training. As such, we advocate the use of this kind of data augmentation when applying deep learning methods to point clouds.

Part	Development computer	Computation machine 1	Computation machine 2
CPU	Intel® Core™ i7-6700HQ CPU @2.60GHz	Intel(R) Xeon(R) W-2125 CPU @ 4.00GHz	AMD Ryzen 7 PRO 5845 8-Core Processor 3.40 GHz
GPU	NVIDIA QUADRO M3000	NVIDIA Quadro RTX 5000	NVIDIA GeForce RTX 3080
RAM	16Go	32Go	64Go

Table 3.6: Hardware configuration used in the thesis.

Appendix D.

3.4.2 Segmentation network

If not otherwise stated, a unique segmentation network is considered in the experiments. This segmentation network is a fully connected neural network using a non-deformable KPConv kernel. It is described as KP-FCNN in the article presenting the KPConv kernel [92]. It was chosen due to its performance (Tab. 2.1), its robustness towards translation [61], its strong spatial awareness but also the convolutive nature of its kernel, which could allow for the creation of a fully-convolutional neural network. The implementation of KP-FCNN used is based on the Torch-Points3D framework [25]. In this thesis, this network is designed by its kernel name: KPConv.

The original method [92] uses an interesting inference scheme. It ascribes a potential value, initially zero, to each point. Each time a point is seen by the network, its potential value is increased. The point of minimum potential values are picked as the centre of the spheres that KPConv uses as input. This process continues until all points reached a fixed potential value. A vote, which consist of using the mean predicted probability for each point, is used to determine the semantic segmentation. This method is not kept for two reasons:

- This voting scheme takes too long a time to segment a complete scene. Initial experiments on the code made by the original method of the author showed an inference time of 20 minutes on Unit 2. This is too far from the target defined initially in Section 1.3 by an order of magnitude.

- It eases the analysis of test results. This voting strategy is not extensively tested, as such its potential advantages and disadvantages are not known.

However, a direct effect of this choice is a decrease in segmentation performance when trying to reproduce the paper performance (an 11.8mIoU points difference on the S3DIS[9] dataset when testing on Area-5 as per the tests performed in Section 5.2.2).

This network is trained for 300 epochs with a batch size of 16. Each data chunk contains at most 4096 points. Preliminary experiments led to using an initial learning rate of 0.005 and a first downsampling of 0.03 m radius when working on the SMARI dataset. When experiments are carried on the S3DIS dataset, the configurations of the original paper (initial learning rate of 0.001 and first downsampling of 0.04 m radius) are used.

3.4.3 Metrics

To evaluate the different results presented in this thesis, the conventional metrics defined in Section 2.2.2, accuracy and IoU, will be used. When training or testing time is reported, the ID of the machine will be specified, as machine 2 is quicker than machine 1.

Unless otherwise noted, the accuracy and IoU metrics presented are a mean between the result obtained by three identical networks trained independently. During the testing phase, inference is carried out ten times, with possibly different points sampled from the data chunk, and a vote is applied between each run to determine the final class of each point. Preliminary tests showed that variation between each inference result is negligible. As such, only the point cloud from the last inference is kept for illustration. If an analysis must be made separately on each scene of the testing dataset, results are computed on this last inference. When not specified, illustrations are made from the best performing network results.

When a mean value is not computed on every class a network is trained on, a "*" symbol will be used to emphasize this difference. The ignored classes will be provided on a case-by-case basis.

For the SMARI dataset, the IoU of some classes will not be presented as these classes are not recognized by the network, such as the electrical components. If not specified otherwise, the reported mIoU and accuracy will nonetheless both be computed on every class.

Chapter 4

Synthetic data

This chapter details the synthetic data generation process devised in this thesis. It is able to work with scenes made of 3D meshes coming from a variety of origin (CAD models, manually made or automatically generated) and transform those meshes in point clouds. To reduce domain shift, this process uses a Virtual Laser Sampling (VLS) method based on a novel virtual laser positioning method. A colouring method used to transfer colour information from unlabelled point clouds to colourless synthetic point clouds is also presented. From the experiments carried out on this synthetic data generation process, synthetic data augmentation and VLS are validated on both a dataset of the literature, S3DIS[10], and our industrial dataset. The results on the colouring method are more nuanced. This method does improve semantic segmentation performance by colouring synthetic data, but only when the colouring of the whole point cloud data is similar. When data is acquired from different location, the improvement brought forth by this method is marginal. The experiments answer RQ1: yes, synthetic data can be processed in a way to reduce domain gap.

The SMARI dataset comprised of acquired data is several times smaller than other datasets of the literature. Use of synthetic data thus appears as a promising way to bridge this gap. The process used to generate synthetic data is presented in Section 4.1. Following the presentation of this process, the use of synthetic data is extensively tested. The influence of the topology of synthetic data is first considered in Section 4.2. This aspect can be considered at three scales: the scene structure, the object model and the local point structure. The exploration of these synthetic data properties is used to test parts of the generation process. A second experimental part, Section 4.3, focuses on the colouring operation applied to synthetic data. Lastly, a conclusion is made following these experiments on the use of synthetic data in Section 4.4.

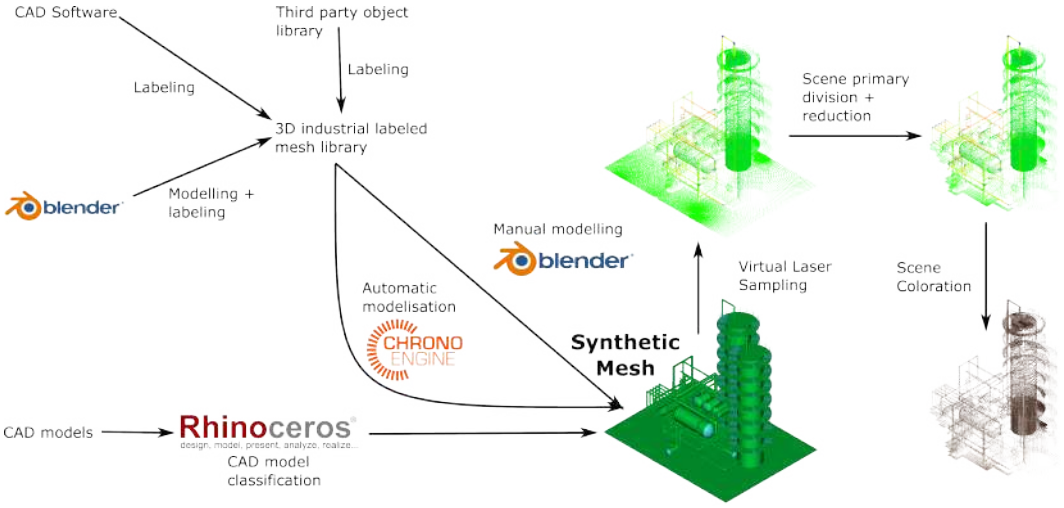


Figure 4.1: Synthetic data generation and processing workflow proposed. The particular software used are presented as references, however, they could be replaced by any software of similar capabilities.

4.1 Synthetic data generation process

The synthetic data generation process, illustrated in Figure 4.1, can be divided in three steps:

1. Create the synthetic scene as a mesh. (Sec. 4.1.1)
2. Transform this scene as a point cloud and optionally reduce this point cloud. (Sec. 4.1.2)
3. Colour the synthetic point cloud.(Sec. 4.1.3)

4.1.1 Generating synthetic scene as mesh

This first step focuses on creating a mesh of an industrial scene and is considered complete when a scene is generated and each of its parts has been annotated. This process varies greatly depending on its input source and the final result can thus be presented as a single mesh file constituted of several sub-meshes or a collection of mesh files already positioned in the same space. A standardisation of the format is carried out during the next step of mesh sampling.

Two pathways of synthetic data generation are considered: using existing CAD models of complete scenes or creating a new scene from individual object meshes. CAD models offer scenes with a high fidelity towards scene structure but may lack details or use 3D models which are fairly unrealistic. Creating new scenes offers flexibility in the scene structure and object representation but needs more work or the creation of an automatic process. When creating a scene from a mesh object library, it was performed manually for industrial scenes. An automatic process was developed to generate office-like scenes in [23].

When creating synthetic data from CAD models, the files were obtained from past works of industrial draughtsmen from Segula Technologies. Those models are often decomposed following the involved disciplines. Each of those models are then constructed in several layers describing different object categories. Thus, the first phase of the work involves reconstructing a complete model from its decomposition. New layers are then used to organise each object following the chosen dataset classification. When the original CAD models are well-made, this process can be done in less than an hour for a hundreds-square-meters scene. As import and export capabilities of each CAD modelling software vary greatly, Rhinoceros[1] was used as it supports a wide array of file formats.

In the case of a created scene, the manual creation process was made in Blender[2], a 3D modelling software. In this case, object are manually positioned in the scene and labelled in the software. The mesh object can originate from CAD models or be manually created in the modelling software. A third option could involve using models already made by third party professionals.

The automatic scene creation process for an office-like scene is based on using a physics-engine [89] and inspired by SceneNet RGB-D [65]. Contrary to SceneNet RGB-D, the method employs a multi-stage strategy of object positioning. An increase in friction forces is also used to diminish a recurrent bouncing effect observed on [65]. This presents the advantage of greatly speeding up computation time and decreasing the risk of badly positioned objects. However, this increase in friction means that objects can sometimes be positioned at a small but unnatural angle (i.e. having only one foot placed on the ground). The detailed algorithm is as follows:

1. A layout is chosen randomly. This layout is a 3D model which defines the scene structure and contains at least the floor, walls and ceiling of the scene. Doors, windows and miscellaneous objects can also be integrated in the layout. At this step, the layout is not added to the physical simulation and only its reference and dimensions are saved.
2. A first wave of objects is added to the scene. Only large objects are considered at this stage. Their initial position is random but bounded by the layout. They are rotated randomly along the vertical axis. Objects are divided in classes following the task at hand. Objects are drawn randomly following their class frequency and objects of the same class have the same probability to be selected.
3. After a sufficient number of simulation iterations to allow the settlement of most objects in the scene, objects which are considered badly positioned by the simulation are removed. This consists of objects which are stuck together or got out of the scene.
4. The second wave of objects is added, this time smaller and placed on the remaining objects of the first wave.

1. https://www.rhino3d.com
2. https://www.blender.org

5. A second phase of object removal is done after the settling of the objects.
6. The layout is properly added to the physical simulation.
7. A third phase of object removal is done after the objects settled.
8. Small defects, such as unnatural angles in object orientation due to the increased friction, can be corrected in this last step.

This method creates physically plausible synthetic mesh scenes. However, objects in such scenes can be positioned unnaturally.

Such synthetic mesh scenes creation methods, from CAD models or objects libraries, manual or automatic, allow for quick data generation but possess two disadvantages. Firstly, the generated scene contains only meshes without textures, which are absent from both CAD models and handmade models. The goal being to generate new data quickly, manually adding textures to those objects is not effective. Secondly, the generated scene still contains over-represented classes due to their realistic organisation. Those disadvantages are partially mitigated by the remainder of the generation process and the reduction step.

4.1.2 Mesh sampling

Once a synthetic mesh scene is generated, it must be transformed into a point cloud in order to be consumed by the segmentation network. This can be done by a process coined "Virtual Laser Sampling" (VLS).

The goal of this process is to simulate a laser acquisition of the scene in order to imitate some flaws inherent to acquired data. This process is divided in two steps, laser position computation and laser sampling. Efforts were focused on laser position computation, the laser sampling is carried out by the Helios++ framework [99]. At first, a custom tool for laser sampling based on ray tracing was used. Its computing time was an order of magnitude greater than Helios++. It was also based on ray tracing and not full-wave simulation, a difference which created a less realistic noise. For these two reasons and motivated by the findings of Noichl et al.[69], the custom ray-tracing method was dropped in favour of Helios++.

The computation of laser position is made based on the following assumptions:

1. The scene is sufficiently complex that information can only be partially acquired from the scene border.
2. Lasers positioned in physically impossible places decrease the point cloud realism. This includes lasers positioned inside objects or floating in the air.
3. Few lasers are needed to gain a sufficient amount of information about the scene.
4. Laser positions visible by other lasers are redundant.
5. Zones with a high density of objects are more informative.
6. Laser positions should be the farthest possible from each other to acquire the most

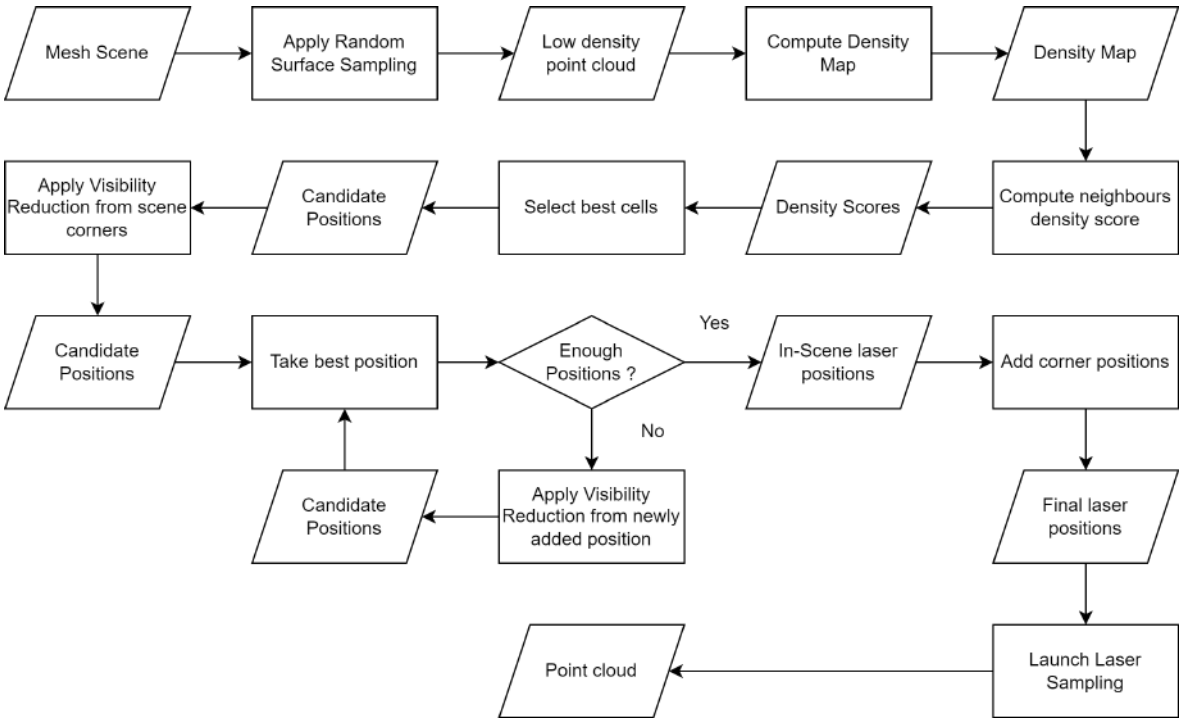


Figure 4.2: Algorithm used to compute the position of virtual lasers.

information possible from the scene.

The computation process is presented Figure 4.2. This algorithm is included in the paper accepted at the SPIE 2023 conference.

Algorithm 1 Adding a mesh to the density map

Require: Mesh M, Density Map D
Ensure: $D.Length > 0$, $D.Width > 0$
  if $M.class \neq sol$ **then**
    $P \leftarrow RSS(M)$
    for all p **in** P **do**
      $D.Add(p)$
    end for
  end if

The initial Random Surface Sampling converts the mesh to a point cloud and a density map of this point cloud is created (Alg. 1). This map is structured as a grid of density. A cell density is computed as the sum of points contained within. As a random surface sampling is used to generate this map, a risk of variation in the created density map exists. However, the number of sampled points used is sufficiently large in practice that no variation was observed in the final laser positions. In the case of a 2D grid following the floor, this process creates cells of high density near highly vertical installation. The number of points sampled from each object can be parameterised following the mesh volume. For this process the floor class is ignored, which implies a density of zero for every cell without object of another class. Each of those zero density cells can be considered as

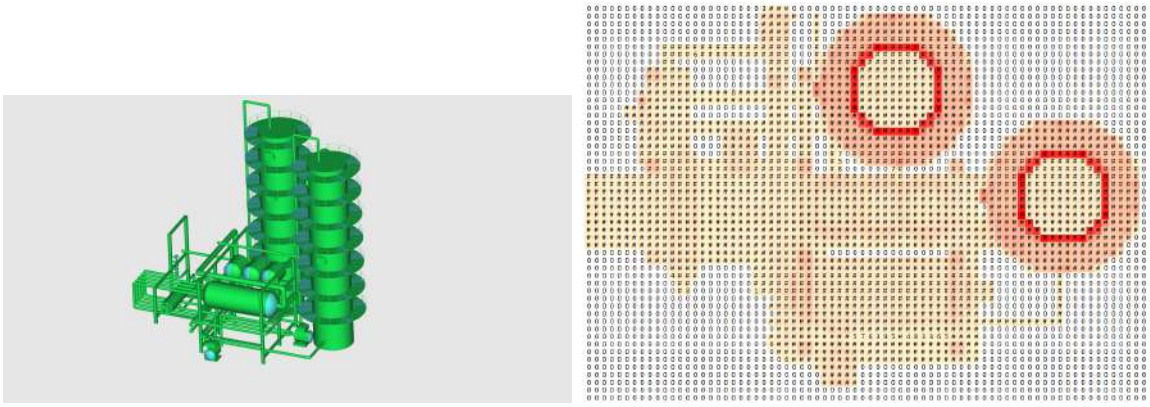


Figure 4.3: On the left, synthetic mesh scene made manually. On the right is the corresponding density map computed on the initial stages of the algorithm.

a potential laser position candidate. Such a density map is illustrated in Figure 4.3.

A density score is then computed for each of those candidates by summing the density of its neighbouring cells. The cells containing the higher scores are selected before proceeding to the next step. An operation called "Visibility Reduction" is then applied to those candidates.

The Visibility Reduction can be defined as follows:

***Definition* 8** (Visibility Reduction)**.** Considering a single cell, called viewpoint, candidate cells can be removed if they are visible from the viewpoint cell.

A cell is visible from the viewpoint cell if the line passing by this cell and the point of view cell only contains cells of density zero. □

The goal of this reduction is to remove any position considered as redundant. Applying this reduction from the viewpoint of scenes corners forces the selection of positions offering new information. Even when considering a 2D grid, where positions which could be considered as invisible from one another by the algorithm are not in reality, this reduction forces the creation of multiple viewpoints around zones which are dense in objects.

The final positions are then selected from the remaining candidates with the help of a greedy algorithm. Its goal is to take the set of possible positions that are the farthest apart. At the first step, the position with the highest density score is chosen. Then, until the desired number of position is reached, the farthest position from the ones already selected is chosen. At each of those steps, a visibility reduction from the point of view of the newly selected cell is applied to the candidates.

Finally, the scene corners are added to this selection of laser positions. These positions are then elevated by one meter above ground and laser sampling can be done thanks to the Helios++ framework, which provides a full-wave simulation of the acquisition process. The virtual laser used tries to imitate the Faro M 70, a medium-range laser. An example of a scene obtained by this process is presented in Figure 4.4.

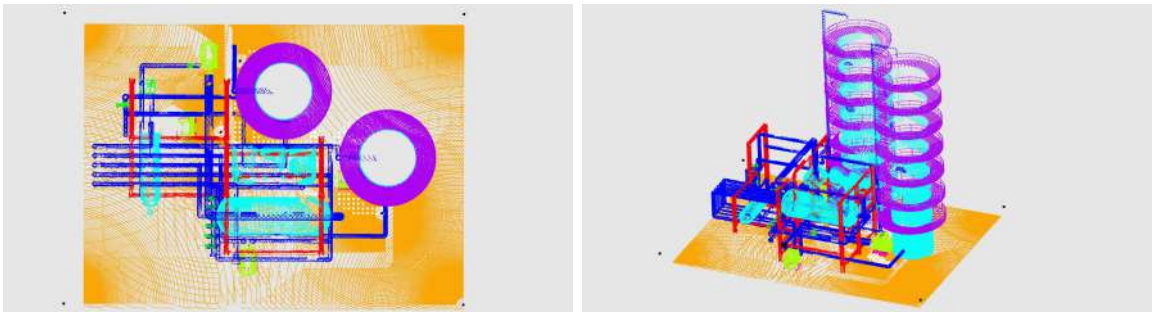


Figure 4.4: Point cloud obtained by VLS from the synthetic scene represented in Figure 4.3. Candidate positions after visibility reduction from the corners points of view are represented as white cubes. The 7 final laser positions are black cubes.

4.1.3 Colouring process

Colour is an important feature in computer vision. In 2D images, they are the primary source of information as pixels are regularly placed colours in a grid. In the case of point clouds, colours can be relegated to a secondary role as the position of points is often enough to determine the function of an object. For example, methods trained on SemanticKITTI [14] offer good results whereas colour information is absent in this dataset. Preliminary experiments also showed that absence of colour during training does not necessarily impede segmentation performance during inference. However, absence of colour data during inference only will seriously decrease performance, which shows that the segmentation network can learn from and rely on this information [23].

When considering industrial data, it is easier to collect point cloud data for an insider than to process it in a semantic segmentation dataset. The colouring process is an attempt to leverage potentially interesting information contained in unlabelled point clouds. In most work focusing in creating synthetic data, the problem of colouring is solved by using predefined texturing. While interesting, this is problematic in our case as no texture map information is defined for CAD models.

The idea is to train a network in recognising recurrent geometric and colour patterns in a point cloud and see how they are related. Thus, the network becomes able to associate interesting colour information to geometric patterns. A generic colour information can be associated in cases where geometric patterns are unusual or not tied to specific colour patterns.

To achieve this goal, a network following the encoder-decoded architecture is conceived. The encoder is based on the non-deformable version of the KPConv kernel (cf eq.(1) and eq.(4) of [92]) and, except for the first, organises each of its layer as a ResNet block (cf figure 2 of [44]). Leaky ReLU is used as an activation function for each deformable KPConv layer. The decoder use a KNN interpolation and MLP to propagate features, as defined in PointNet++ [76]. Skip links are used between the encoder and the decoder to convey information for the upsampling steps. A last layer composed of an MLP and

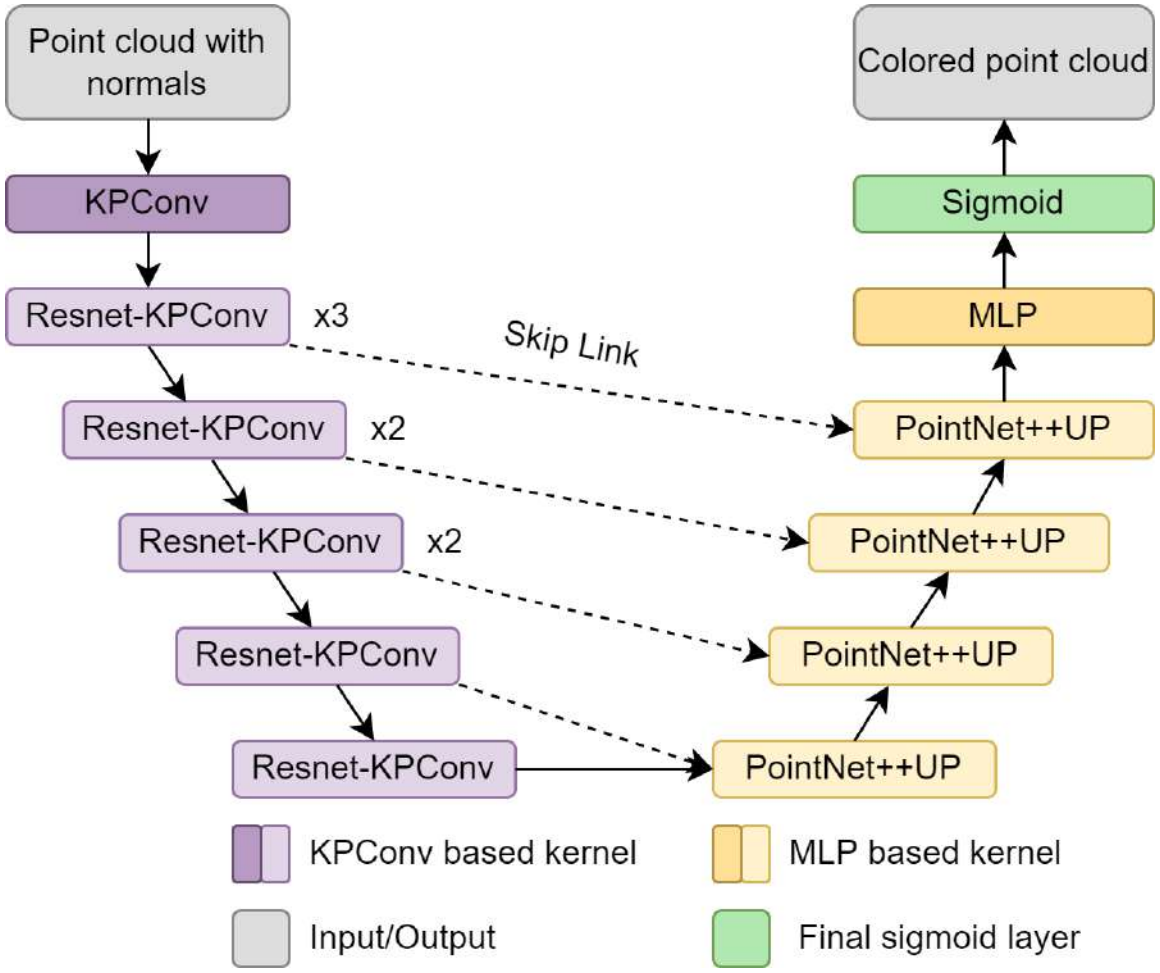


Figure 4.5: Colouring network used

a Sigmoid is used to determine the colour of each point. The network is illustrated Figure 4.5.

The colouring problem can be considered as a regression problem. As such, the Mean-Square Error (MSE) is used as a loss to train this network. When considering a point cloud P of size N as defined in Definition 1, the real colour features vector Fc of P and the predicted colour features vector $\widehat{Fc}$, this loss can be expressed as:

$$MSE_{loss} = \frac{\sum_{i=1}^{N} ||\widehat{fc_i} - fc_i||_2}{N} \tag{4.1}$$

Using the mean square error as a loss allows the network to focus on highly specific patterns and to ignore unspecific zone. This is due to the tendency of MSE to penalise strongly large errors and reduce the impact of smaller ones. This comportment is suitable for the envisioned goal behind the colouring network creation. This loss can also be used on any colour format such as RGB (Red-Green-Blue) or HSV (Hue-Saturation-Value) without modifying the network structure, as long as the data format used is identical for the same set of network weight.

However, this loss does not guarantee that the network will use the whole colour

spectrum. Preliminary tests also showed its tendency to concentrate the spectrum around a mean value, which is to be expected due to the loss nature. Thus, two regularisation functions are evaluated.

The first one penalises the lack of use of the darkest and brightest colours present in the training data. We call this the distance regularisation R_{dist}. When considering the real colour spectrum of the point cloud S and its predicted counterpart $\widehat{S}$, R_{dist} can be defined as:

$$R_{dist} = ||min(S) - min(\widehat{S})||_1 + ||max(S) - max(\widehat{S})||_1 \tag{4.2}$$

This regularisation presents the advantage of being simple, straightforward and quick to compute. However, it can be easily gamed by the network as it only considers the spectrum extrema. This can lead to a network focusing mostly on the centre of the spectrum and only marginally using the extrema to comply with the regularisation.

The second regularisation is based on the variance of the colour spectrums. It adds the difference between the real and predicted variances to force the network to use a more diversified colouring. This variance regularisation, R_{var}, is defined as:

$$R_{var} = ||\sigma^2(S) - \sigma^2(\widehat{S})||_1 \tag{4.3}$$

R_{var} presents the advantages of focusing on colour diversity more than colour extreme values. However, this forced diversity can come at the cost of an informativeness loss in the point cloud colours. Colour is spread more gradually between points and the edge between different regions of the point cloud becomes blurry.

When not specified, an HSV colour representation will be used to train the colouring network. The idea behind this choice is that colour information is better understood in this manner when working with acquired data. For example, the acquired colour of a pipe can vary due to lightning even when it is uniformly painted. HSV, with its concepts of hue and value, will handle this discrepancy better by design than RGB. This is due to value representing in HSV the intensity with which light is shone on an object. This choice is furthermore justified by 2D colourization patch based methods which can transfer colour from RGB images to grey-scale images with only the help of luminance information [16].

4.2 Topology of synthetic data

The point clouds considered in this thesis are 3D representations of our environment. As such, their topology can be considered. In this part, we will use a somewhat informal definition of topology adapted to point cloud: “How does objects composing a point cloud relate spatially to each other?”. When observing a point cloud, three levels of object size become apparent: the layout of the whole scene (Sec. 4.2.1), objects shape (Sec. 4.2.2)

and the local point structure (Sec. 4.2.3). By taking those topological levels of data representation in consideration, several experiments can be designed to test the synthetic data generation process. Those tests will focus on the synthetic data generation process as well as the sampling process.

4.2.1 Global scene layout

When looking at the topology of a scene, the highest level that can be considered is its general structure. The general layout of a scene matter in its understanding as it define the relationship between object and possibly their meaning. Sometimes, different objects are indistinguishable from their shape alone and context is the defining factor. For example, electric cables and small bore pipes are nearly identical in their geometry [6].

The goal of this section is to look into the part played by a synthetic scene layout in its informativeness. Even if the scene is divided in several parts later in our proposed data processing workflow, the general layout of a synthetic scene influences its sampling process. This influence also leads to a different object representation due to occlusions in the virtual acquisition process. Finally, the contextual informativeness brought by a realistic layout should not be forgotten.

To test these hypotheses, the following experiment will be carried out:

1. Selecting a test scene from the acquired dataset.
2. Creating a synthetic scene which imitates some characteristics of the chosen test scene.
3. Adding the created synthetic point cloud to the acquired training set. Training and testing following the general methodology are carried out on this first modified dataset.
4. Shifting randomly every object, except the floor, in the synthetic scene in order to disorganise it. Objects are shifted in the three dimensions. Objects going out of the scene on one side are transported back to the other end of the scene in order to roughly keep the same scene size.
5. Adding the disorganised synthetic point cloud to the acquired training set. Training and testing following the general methodology are carried out on this second modified dataset.
6. Shifting randomly the floor along the vertical axis in the disorganised scene.
7. Adding the even more disorganised synthetic point cloud to the acquired training set. Training and testing following the general methodology are carried out on this third modified dataset.

Two different scenes are picked to carry out this experiment, the Piping and Plan scene, presented in Figure 4.6. The synthetic scenes are manually modelled and shifted. The

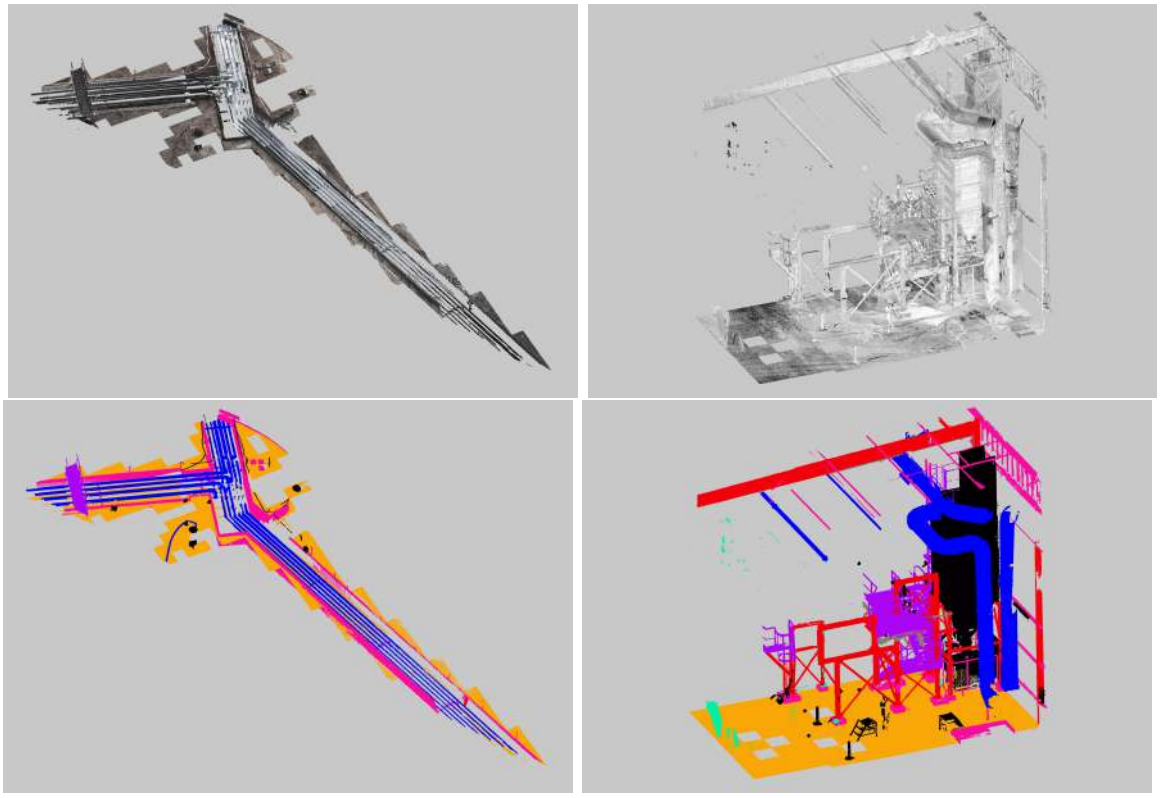


Figure 4.6: The Piping (left) and Plant (right) scenes, in raw format on top and their manual segmentation at the bottom.

synthetic piping scene focuses on recreating the trench aspect of the original scene. The synthetic plant mimics the large HVAC piping, which represent a problem during segmentation when only acquired data are considered. Their different versions are presented in Figure 4.7.

The first observation that can be made following the results obtained from this experiment (Tab. 4.1) is that using synthetic data is helpful to increase segmentation results quality. A first look at the classes of interest shows a positive evolution of segmentation performance in most cases. Moreover, the single synthetic scene seems to play its intended part in both cases.

In the Piping case, an increase in performance when segmenting the pipe, floor and structure classes is visible. Notably, the IoU for the structural class increases by 12.54. As the increase in this class falls sharply when the structure of the synthetic scene is disorganised, this provides a good argument in favour of our hypothesis: layout influences segmentation performance.

The synthetic Plant scene was less similar to the Plant test scene and was mostly trying to correct problems with HVAC objects. This lead to a lesser increase in general IoU but is still the best performing case on the pipe and beam classes.

In each case, disorganising the scene lessen segmentation performance, with even a case where adding synthetic data slightly decreases the network ability (-0.42 mIoU for the

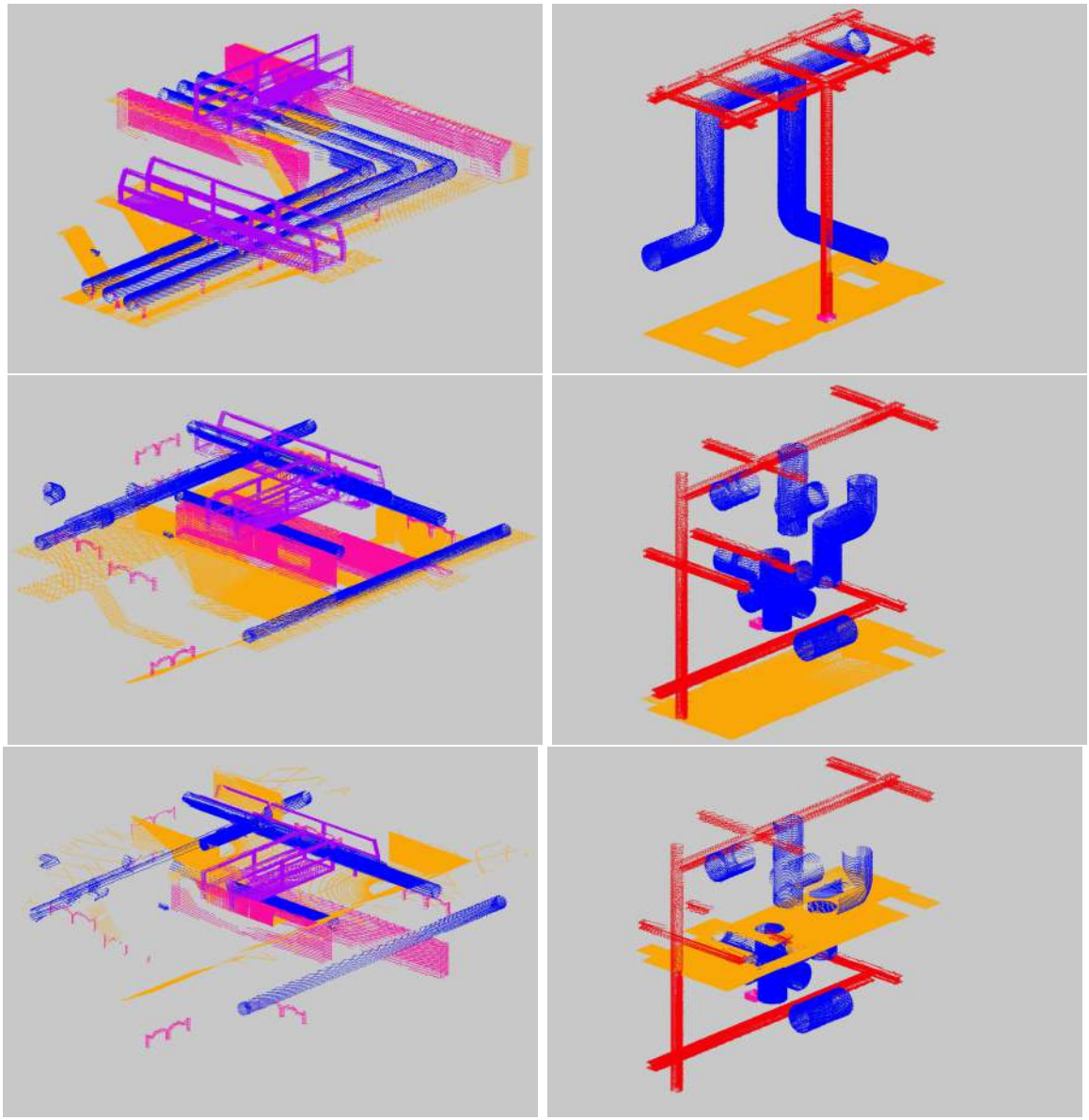


Figure 4.7: Synthetic data manually generated for the experiment with the Piping (left) and Plant (right) test scenes. From top to bottom: as generated, with shifted objects, with shifted objects and floor.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	69.30	18.82	66.67	0.27	5.76	32.29	82.55	42.42	28.11
Piping	**72.61**	**20.77**	68.25	1.37	7.73	38.75	**83.96**	41.82	**40.65**
Piping shifted	71.84	19.82	68.02	0.20	3.38	**38.76**	81.76	43.92	37.17
Piping floor shifted	68.70	18.40	66.49	0.78	2.78	33.88	81.22	40.57	26.51
Plant	70.73	19.49	**69.09**	**6.04**	3.38	34.37	82.97	46.06	28.28
Plant shifted	67.64	18.79	60.28	3.07	11.40	33.73	79.43	45.60	24.91
Plant floor shifted	68.76	19.18	61.50	1.44	**12.07**	36.47	81.98	**47.83**	22.93

Table 4.1: Results of the synthetic data scene layout experiment.

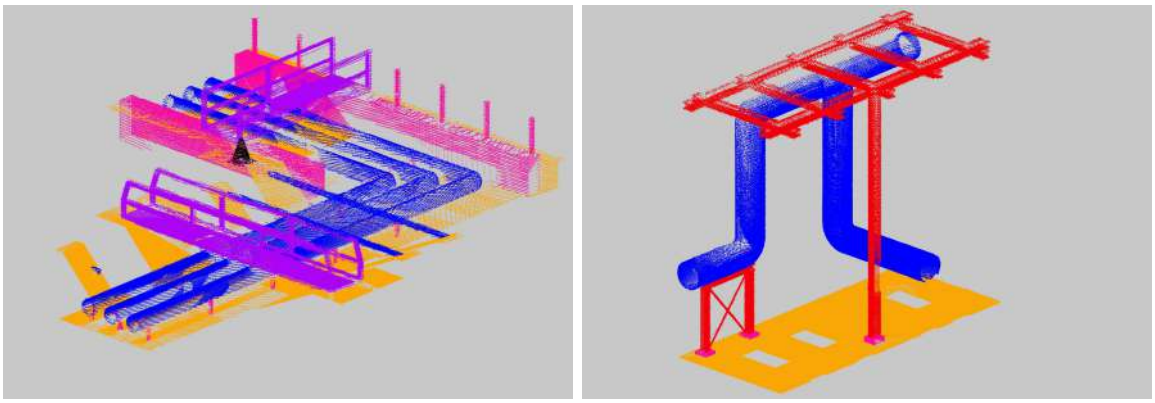


Figure 4.8: Manually enhanced synthetic data generated for the experiment with the Piping (left) and Plant (right) test scenes.

piping scene with the floor shifted compare to acquired only data). When looking at more fine-grained detail, the relationship between object representation and scene structure is more apparent. A decomposition of the piping system in the Plant case sharply decrease the network ability to segment this class (-8.81 and 7.59 IoU). The separation of the walls and the floor in the piping scene brings a similar decrease, much more prominent when the wall class is totally disconnected from the floor in the third case (-14.14 IoU of structural class). This shows the segmentation network ability to understand basic relationship between classes of objects.

Layout has a clear influence on the network segmentation abilities. Is it possible to further enhance our handcrafted synthetic scenes by adding other objects similar to those present in the originals? Additional pipes and structural elements are added to the Piping cases. The structural elements mimic the ones holding a chain along the trench in the original scene. For the Plant case, a single T-brace is added under a pipe, which is an important element missing from the reconstitution. Both enhanced scenes are presented in Figure 4.8.

This manual enhancement fails in the Piping scene case, where performance decreases sightly. It is however a success in the second case (Tab. 4.2).

The slight decrease in the Piping scene case (-0.9 mIoU) seems to be due to the

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	69.30	18.82	66.67	0.27	5.76	32.29	82.55	42.42	28.11
Piping	**72.61**	**20.77**	68.25	1.37	**7.73**	38.75	83.96	41.82	**40.65**
Piping enhanced	72.23	19.87	**69.21**	0.77	2.12	**39.23**	**84.54**	39.48	38.61
Plant	70.73	19.49	69.09	6.04	3.38	34.37	82.97	46.06	28.28
Plan enhanced	72.02	20.37	70.14	**7.52**	5.01	36.80	83.70	**46.68**	32.65

Table 4.2: Results on the enhanced synthetic data scene layout experiment.

interference of the new structural elements. Those hollows are rectangular shaped and quite large, which differ from the ones in the original, rounder and thinner. Moreover, their shape is similar to the handrail support in the walkways of this synthetic scene. The decrease in segmentation performance for both of those classes (-2.34 and -2.04 IoU on walkway and structural, respectively) seems to confirm this hypothesis.

Adding 7 elements to the synthetic Plant scene increased mIoU by 0.88. This slight increase is observable in every class, most notably on the beam class (+1.48 IoU). This can be attributed to two things. First, by adding possible context surrounding the pipe class. Second, no T-Brace are present in the trained part of the acquired dataset. This increase in performance shows the network ability to gain information from small, well-made, data addition.

Both these experiments show the importance of well-made and structured synthetic data. However, they also seem to hint at the importance of individual object shape as a realism factor. Thus, this aspect is the one studied in the next section.

4.2.2 Synthetic object realism

To study the influence of individual synthetic object shape, a focus on the valve class is made. This class was chosen as it is important but poorly segmented with the acquired-only dataset and represented by a high variety of shape (Sec. 3.1). Two cases of study are presented. The first one focus on a manually constructed scene containing a high number of valve model. The second uses a crude point cloud generation method to balance the dataset in favour of pipes and valves.

4.2.2.1 Manual scene construction

A synthetic scene containing a high number of valves is constructed manually and added to the acquired-only dataset. Two versions of this scene exist. The first one uses 3D models of valves which are semi-realistic (Fig. 4.9), the second one contains schematic models of valves as used in CAD software (Fig. 4.10).

The test results obtained after training a segmentation network on these datasets are quite surprising as the Schematic case obtains better results on the valve class (Tab. 4.3). However, the scene with semi-realistic models is better or equivalent than it in segmenting

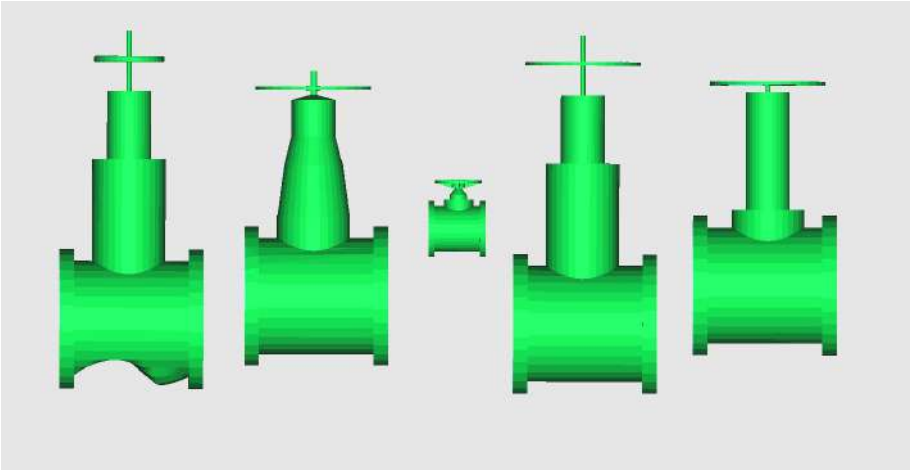


Figure 4.9: Examples of semis-realistic valves models. The principal components of valves are represented (flange, shaft, wheel) but the shape of the shaft is simplified.

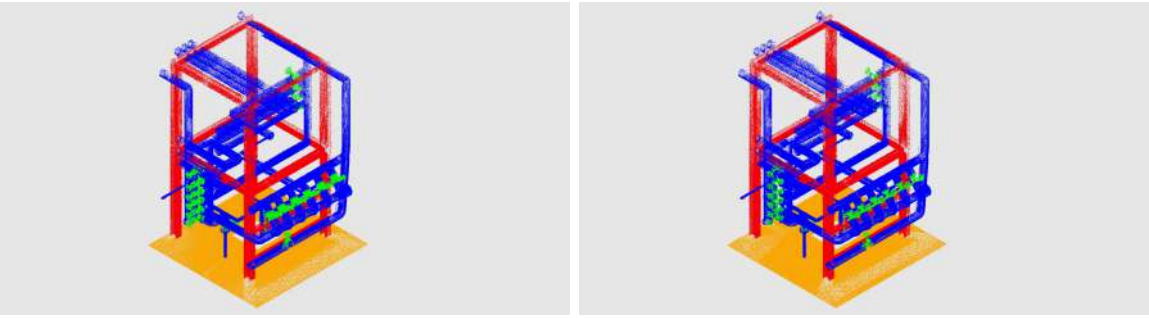


Figure 4.10: Manually create scene with a focus on the valve class, with semi-realistic model (left) and schematic model (right).

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	69.30	18.82	66.67	0.27	5.76	32.29	82.55	42.42	28.11
Semi-Realistic	**71.55**	**20.09**	**70.01**	**4.98**	4.36	**37.61**	**84.11**	42.80	**31.05**
Schematic	70.30	19.30	66.99	3.70	**5.98**	35.17	82.77	**42.84**	29.54

Table 4.3: Results on a synthetic scene with either semi-realistic or schematic valve models.

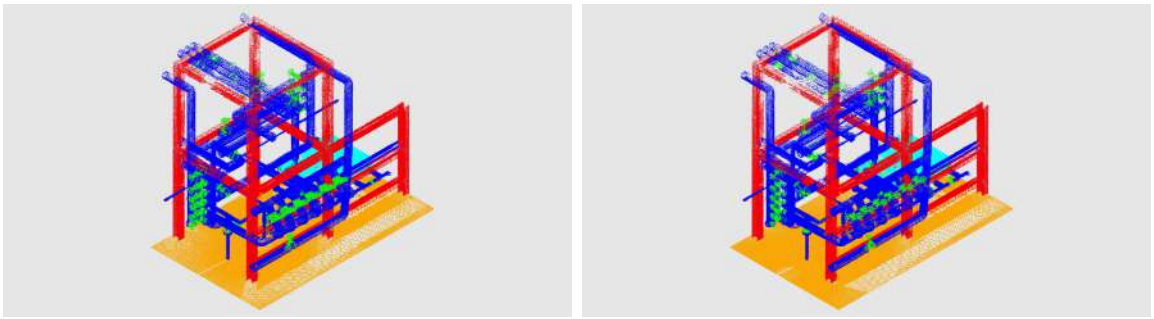


Figure 4.11: Expanded scenes, with semi realistic valve models (left) and schematic models (right).

every class except for valves. This result seems to imply that the schematic valve contains easier features to learn compared to the more realistic model. Thus, the network training goes in different directions by following a path of less resistance depending on the data:

- One focuses on the valve but does not succeed in going much further than the acquired-only dataset in its valve-segmentation capability due to the schematic nature of the model (+0.22 IoU relative to acquired-only).
- The other has a more holistic approach but focuses less on the valve class and thus see its valve-segmentation capability decreasing (-1.40 IoU relative to acquired-only).

For this experiment, only a few different models of valve were used, albeit at three different scales, but without variation between identical models. Would using a higher variety of valve model be useful, even if they were still only semi-realistic? To answer this question, the synthetic scene is expanded and a greater variety of valve model is used. This is done by using different models but also adding variation to the same model, such as rotating the valve wheel or moving it along its axis (Fig. 4.11).

As before, the segmentation model is trained on two versions of the scene: One with semi-realistic models and one with the schematic model only. As seen in Table 4.4, expanding the scene fails to improve the segmentation performance on the valve class. Worse, expanding the scene only slightly improves results on the schematic case and degrades them on the semi-realistic case.

Does expanding the scene in this case scatters the network attention during training? This dispersion of attention would then create a network that focus on easy to get features (schematic valve, beam) instead of more diverse ones (semi-realistic valve). To answer this

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	69.30	18.82	66.67	0.27	5.76	32.29	82.55	42.42	28.11
Semi-Realistic	**71.55**	**20.09**	**70.01**	4.98	4.36	**37.61**	84.11	42.80	31.05
Schematic	70.30	19.30	66.99	3.70	**5.98**	35.17	82.77	42.84	29.54
Semi-Realistic expanded	71.23	19.68	67.50	**5.40**	1.65	33.71	83.73	44.90	**34.16**
Schematic expanded	71.04	19.51	68.33	4.02	4.40	33.37	**85.03**	**45.46**	29.90

Table 4.4: Results on the expanded synthetic scene with either semi-realistic or schematic valve models.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	69.30	18.82	66.67	0.27	5.76	32.29	82.55	42.42	28.11
Semi-Realistic	**71.55**	**20.09**	**70.01**	4.98	4.36	**37.61**	**84.11**	42.80	31.05
Semi-Realistic expanded	71.23	19.68	67.50	**5.40**	1.65	33.71	83.73	44.90	**34.16**
Semi-Realistic enhanced	70.94	19.88	69.89	4.16	**5.92**	34.06	83.91	**47.22**	30.32

Table 4.5: Results obtained by enhancing the synthetic scene with semi-realistic valve models.

question, the initial non-expanded semi-realistic scene is enhanced by adding more valve and diversifying their 3D model in the same manner that was applied to the expanded case.

The results obtained when training with this enhanced scene are exposed in Table 4.5. They seem to indicate that by adding more valve information in the dataset for the same quantity of data (in m^2), we could slightly gear the network towards focusing on this class. Networks trained on this scene focus more on the valve class (+ 1.56 IoU on valve class) at the cost of other classes, as pipe, beam, tank, floor and structural are less well segmented.

4.2.2.2 Automatic scene generation

A last effort made towards understanding the effect of synthetic data is to create high quantity of synthetic scenes from acquired point cloud. In order to create such a scene the EIF dataset [107] is used to supply valve and pipe point cloud. A simple scene creation technique is envisioned and described by Algorithm 2 and Algorithm 3.

This technique creates long lines of intertwined pipes and valves. As it requires objects centred around the origin and, if possible, upward-oriented valves, the EIF dataset [107] must be preprocessed. In this dataset, objects are positioned as is defined by the acquisition step of its authors: they are not around the origin nor oriented upward. Centring objects around the origin is trivially done by a translation between the objects centres and the origin. To orient the valves upwards, the basis of each point cloud is computed by using its eigenvectors. The point cloud can then be oriented upward following this basis. This method allows to quickly process every valve of EIF without supervision but

Algorithm 2 Creating a valve-pipe scene.

Require: Uniform Random Number Generator R, List of Pipe Model P, List of Valve Model V

Ensure: $P.Length > 0$, $V.Length > 0$

```
initialX ← R.Draw(−100, 100)
limitX ← initialX + R.Draw(20, 100)
currentY ← R.Draw(−100, 100)
numberOfLine ← R.Draw(1, 6)
for i = 0 to numberOfLine do
    currentX ← initialX
    maxYShift ← 0
    while currentX < limitX do
        pipe ← P[R.Draw(0, P.Lenght − 1)]
        AddObject(currentX, currentY, maxYShift, pipe, 0)
        valve ← V[R.Draw(0, V.Lenght − 1)]
        AddObject(currentX, currentY, maxYShift, valve, 2)
    end while
    currentY ← currentY + maxYShift
end for
```

Algorithm 3 Adding an object to the generated scene.

Require: $currentX$, $currentY$, $maxYShift$, object O, Semantic Class Number C

Ensure: $currentXisreference$, $maxYShiftisreference$, $O.center = (0, 0, 0)$

```
O.Translate(currentX, currentY, 0)
SaveToOutput(O, C)
currentX ← currentX + O.XMax − O.XMin
MaxYShift = Max(MaxYShift, O.YMax − O.YMin)
```

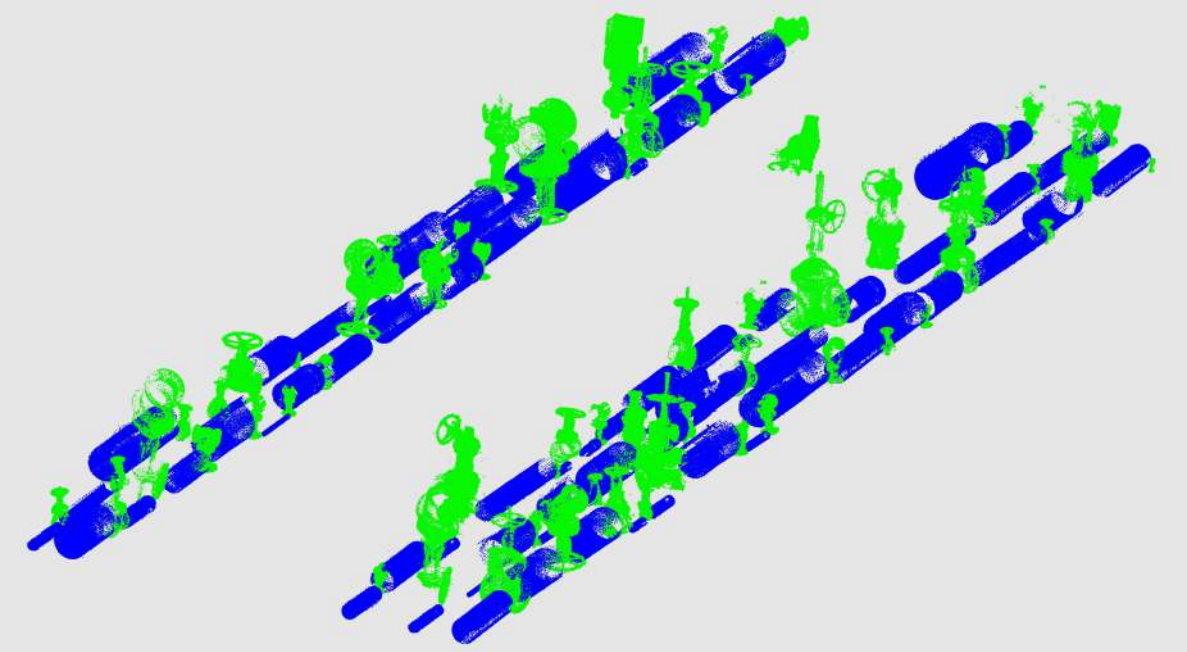


Figure 4.12: Lines of pipes and valves automatically generated with the method presented by Algorithm 2.

does not succeed every time. The point clouds contained in EIF are acquired and thus imperfect representation of the original object. Occlusions occurred during acquisition and the computation of the point cloud eigenvectors only reflects the point cloud basis, not the original object basis. When a significant part of the object is missing, its basis can be oriented along unwanted directions, for example between its wheel and flange and not along the wheel. Moreover, even when the point cloud is relatively well-made, this process does not guarantee an upward-oriented object but only an object oriented along the axis of the canonical basis. To solve this problem automatically, the up direction of a valve is determined to be its longest dimension and the lateral dimension its smallest dimension.

As seen in Figure 4.12, a scene generated by this method contains a high number of pipes and valves which are only roughly organised. Some valves contain outlier points, which create a higher than needed transversal shift. Some valves are oriented upside-down and the bore size between adjacent pipe and valve is not valid. As the scene layout is of poor quality, numerous scene are generated to follow a philosophy often present in the literature: quantity beat quality[65]. This also keeps us in line with the previous experiment, where an increase in the number of point representing valve per square meters seemed to help the network focus on this class.

20 scenes are generated with this method (2900 m^2) and added to the acquired only dataset under the name KAL. Sadly, as reported in Table 4.6, the network performance decreases when presented with this additional data. This is true in general but also when considering the two classes targeted: pipe (-5.65 IoU) and valve (-2.61 IoU). Two causes

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	**69.30**	**18.82**	**66.67**	**0.27**	**5.76**	32.29	**82.55**	**42.42**	28.11
KAL	67.97	18.51	61.02	0.07	3.15	**34.44**	79.93	40.49	**32.68**

Table 4.6: Results after adding automatically generated point cloud to the acquired only dataset

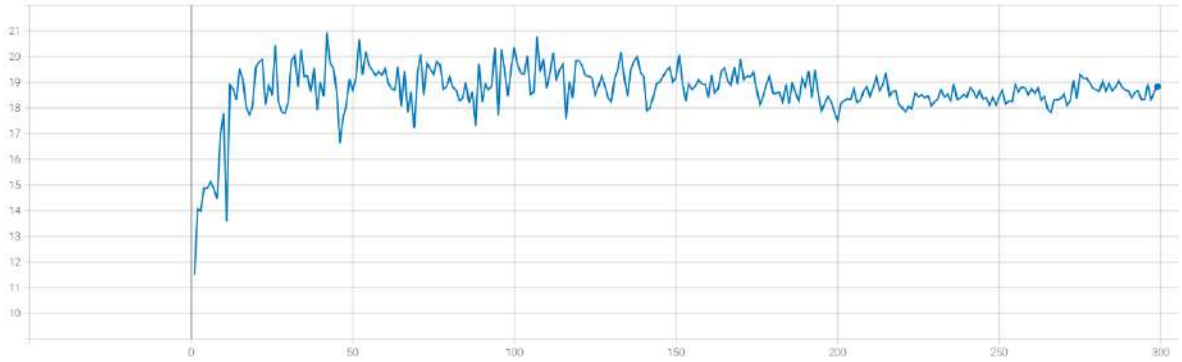


Figure 4.13: mIoU per epoch on the testing set during network training. The network seems to stop learning new knowledge after the first 50 epochs.

can be advanced behind this decrease in segmentation quality.

Firstly, the general structure of the KAL scenes is imperfect. The connection between valves and pipes is often badly done and the orientation of the valves is not properly controlled. As seen in Section 4.2.1, the structure of the scene do matter. This can lead to an increased difficulty for the network to extract knowledge during the learning phase. Secondly, adding so many data whose structure is roughly similar seems to lead to a slight overfitting during training (Fig. 4.13). This also imbalances the dataset with valve and pipe now representing 33.36% and 36.37% of the dataset in terms of number of point respectively.

The results from these experiments are more nuanced than those on the scene layout. It seems that, when focusing on a class that is less represented, such as valves, the quantity of data matters more than its quality. However, this high quantity of data cannot be introduced in the dataset without negative consequences if the previous findings on scene layout are not respected. Thus, when working with classes in a situation similar to the valve class, the primary effort can be focused on reducing class imbalance instead of working on domain shift if the synthetic data is correctly built.

4.2.3 Sampling process

In the literature, when meshes must be transformed to point clouds in the context of deep learning tasks, Random Surface Sampling (RSS) is used (Sec. 2.3.3.1). Only works focusing on synthetic data will consider using other methods (Sec. 2.3.4). One of the problems associated with using RSS is its impact on local point structure. Points are not

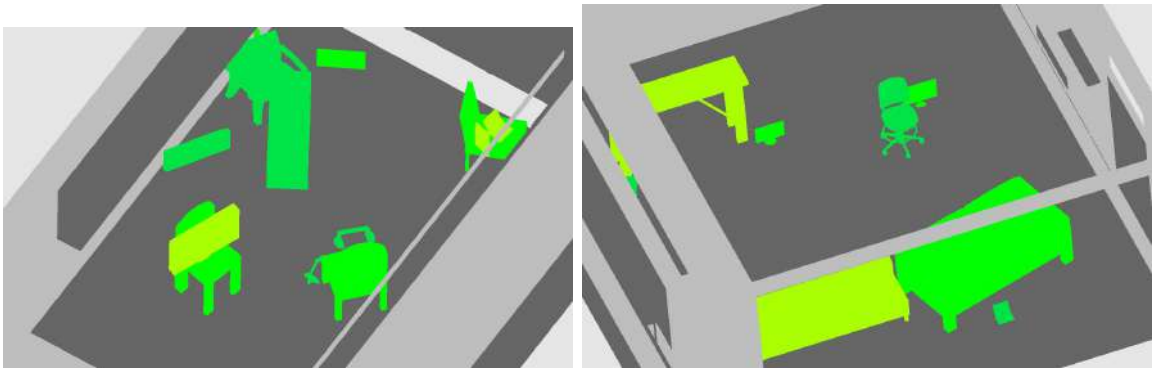


Figure 4.14: Automatically generated mesh scenes.

organised coherently, as it is the case when real acquisition methods are used. This section will thus focus first on studying the different between using VLS and RSS. Following this study, the influence of the virtual laser parametrisation will be considered. Finally, the relevance of the laser positioning algorithm will be validated. The first two items of study correspond to work published in the SMC 2022 conference [23] and are applied to the interior dataset S3DIS [9].

4.2.3.1 S3DIS

Before studying the impact of the sampling process on synthetic data, synthetic mesh scenes must first be generated. The automatic generation process described Section 4.1.1 is used. The layouts used are from SceneNet [43]. Only layouts of office type are used as non-office-like rooms make up for less than 5% of S3DIS [9]. The object library used is the ShapeNet dataset [24] and object size and frequency were estimated from the first zone of S3DIS. A uniform distribution was used for the remaining random variables: choice of layout, number of objects, object position and orientation. Example of generated scenes are presented Figure 4.14.

Training is done on the first zone of S3DIS and 3060 m^2 of synthetic data. The data is divided in pillars with a one square meter base. The 3060 m^2 of synthetic data is obtained after a reduction on pillar containing only the floor, wall and ceiling class. The test is done on S3DIS zone 5 and the zone 2, 3, 4 and 6 are used to train the colouring network. As the generation method cannot create scenes containing the beam and column class, a mIoU* metric which excludes those two classes is also used.

In this experiment, the lasers are not positioned according to the result of the algorithm described Section 4.1.2. Instead, a simpler approach is taken where a laser is positioned depending on the layout. When the layout is composed of only one room, a laser is positioned at its centre and each corner. When the layout is composed of several rooms, two lasers are positioned in each room. They are on the first and second thirds of a line positioned at the centre of the room along its longest side. In any case, lasers positioned

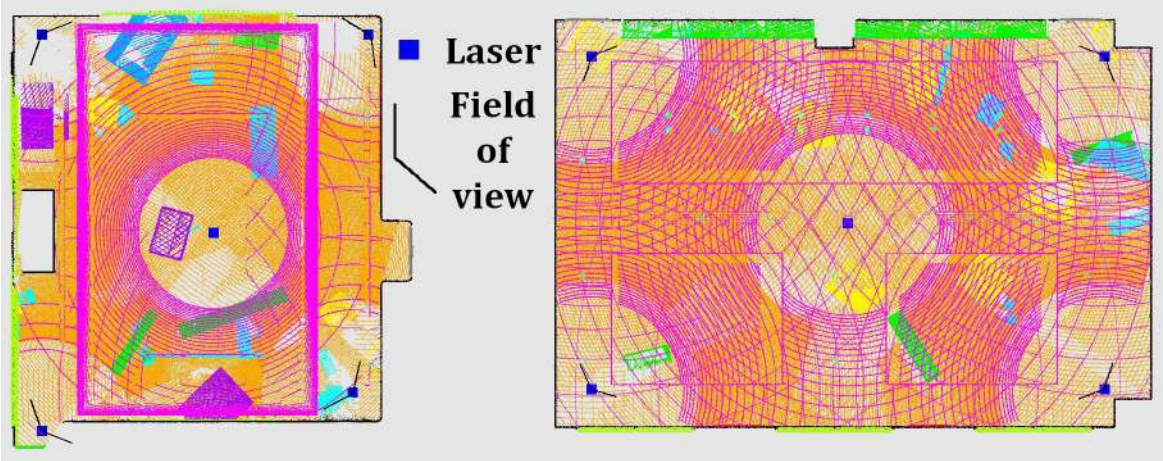


Figure 4.15: Examples of laser configuration used in office-like scenes.

Sampling	OA	mIoU	mIoU*	beam	board	book.	ceil.	chair	clut.	door	floor	table	wall	col.	sofa	wind.
Zone 1 Only	67,6	31,7	39.9	0,1	18,7	44,2	60,7	61,5	37,1	27,7	46,4	53,6	68,0	**5,0**	7,7	13,7
VLS	**71,1**	**34,4**	**43.4**	0,1	**21,9**	44,3	**62,0**	**65,0**	**37,5**	**35,5**	**57,5**	**54,2**	**70,6**	3,8	**9,5**	**19,0**
RSS	69,3	31,6	39.8	0,0	13,0	**44,4**	61,7	61,1	37,1	28,1	53,7	50,7	68,8	3,9	4,4	15,3

Table 4.7: Inference result on S3DIS zone 5, the best result out of three networks is shown.

at the centre have a 360° field of view and lasers in the corners a 130° field of view oriented toward the scene centre. Two examples are presented Figure 4.15.

When changing the sampling method used, a clear difference in favour of VLS appear (Tab. 4.7) with a difference of 3.6 mIoU* and 2.8mIoU. In the case of RSS, adding synthetic data does not really improve performance compared to only using the first zone of S3DIS. The overall accuracy is higher but mIoU is slightly lower. VLS obtains the best result in all but three classes. The superiority of this sampling method for synthetic data augmentation is clear compared to RSS.

Following this assessment, the parametrisation of the virtual laser can be studied. By mimicking the characteristic of existing laser equipments, three configurations are proposed in Table 4.8. The long range configuration was used in the previous experiment. Those long, medium and short ranges imitate the Riegl VZ-400, Faro M 70 and Matterport Pro 2 acquisition devices respectively. The first two are laser device, with the Riegl VZ-400 being a high-end long-range equipment and the Faro M 70 a lower-end medium-range laser. The original Matterport Pro 2 device, which was used during the acquisition of the S3DIS [9] dataset, output meshes. The virtual laser tries to mimic its result once converted to point cloud in the most faithful manner possible.

Laser	Min range (m)	Accuracy (m)	Beam divergence (rad)
Long range	1.5	0.005	0.0003
Medium range	0.6	0.003	0.0042
Short range	0.13	0.01	0.001

Table 4.8: Virtual laser configuration considered.

Virtual laser	OA	mIoU	mIoU*	beam	board	book.	ceil.	chair	clut.	door	floor	table	wall	col.	sofa	wind.
Long range	72.2	33,6	42.5	**0,2**	18.3	42,3	69,6	52,5	34,3	35,8	77,2	55,4	66,2	3,8	**11,2**	4,3
Medium range	72.5	33.6	42.5	0,1	16,1	41,5	70,6	54,6	35,1	**36,9**	77,4	55,4	65,6	3,8	9,3	4,4
Short range	**72.6**	**33.8**	42.5	0,1	14,1	42,1	**71,2**	56,4	35,1	35,2	**78,0**	**56,8**	65,4	4,7	8,7	4,8
Zone 1 Only	67,6	31,7	39.9	0,1	**18,7**	**44,2**	60,7	**61,5**	**37,1**	27,7	46,4	53,6	**68,0**	**5,0**	7,7	**13,7**

Table 4.9: Differences obtained by changing the virtual laser configuration used in VLS.

When looking at the results obtained when only the laser configuration changes during VLS (Tab. 4.9), there is no visible difference in segmentation performances. This is true even when trying to imitate the acquisition process of the original data with the short-range laser. We can thus postulate that, if the local point structure is an important component of realism and a major factor of realism, it is only true until a certain level of detail. The variations created by changing the laser configuration in the local point structure of the cloud are too minute to really influence segmentation performance. The use of farthest point sampling during the pooling phases of the KP-FCNN network is certainly responsible for this effect. This layer favouring the creation of uniformly distributed point clouds, it will ignore points which are too close. Thus, the effect of extremely local structure is limited to the uttermost layers of the network.

Following this observation, the medium range laser will still be used for the SMARI dataset.

4.2.3.2 SMARI

The previous experiments validated the Virtual Sampling Method as it is. It is now possible to check the relevance of the laser positioning algorithm. In order to answer this question, four laser configurations should be tested:

1. Random laser positioning, which randomly draw lateral and longitudinal coordinates. (L_R)
2. Corners only laser positioning. (L_C)
3. Lasers positioned by the algorithm but without visibility reduction. (L_A)
4. Lasers positioned by the algorithm with visibility reduction, as defined in Section 4.1.2. (L_AVR)

In order to perform this experiment, three synthetic scenes are considered. This set of synthetic data is used repeatedly in the thesis experiments and coined Synth3 in following works. Two simple scenes which are simplistic and whose only role is to slightly balance the dataset by including mostly beam-class object. Those two are sampled by VLS with lasers which are positioned by the algorithm. Due to their simplicity, the visibility reduction was not deemed useful and is thus not used. The only variable is how the more complex third synthetic scene is sampled. Except the corner only positioning, each configuration determines three positions. Nine different random laser configurations are used for this experiment.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired only	70.51	19.82	67.97	7.16	2.48	33.74	81.33	50.11	30.91
RSS	66.29	18.30	60.59	10.25	2.31	32.69	77.11	43.78	26.72
Mean L_R	72.35	22.29	70.46	29.21	**4.14**	38.26	81.41	50.90	34.83
L_R_3	**74.35**	23.08	**74.51**	29.05	3.50	41.89	82.63	49.21	39.98
L_R_2	70.76	21.32	65.30	**30.01**	3.64	35.98	78.48	48.56	33.90
L_C	69.06	20.38	57.77	27.63	2.77	35.09	78.85	38.16	42.65
L_A	72.94	22.50	72.21	27.33	4.11	38.64	81.79	51.38	36.83
L_AVR	74.33	**23.39**	73.22	29.90	2.42	**43.35**	**82.93**	**53.02**	**39.89**

Table 4.10: Results obtained by varying the laser configuration used. The mean result of every random laser configuration is provided as well as the best (L_R_3) and worst (L_R_2) results.

The results obtained, Table 4.10, show the prevalence of computing the laser positions with visibility reduction. Its mIoU is superior to any configuration obtained randomly, contrary to a computation without visibility reduction, which is only superior on average. Obtaining information from laser present only on the corner is not enough (-0.94 mIoU compared to the worst random laser configuration). It is nonetheless superior to baseline RSS or acquired data only and thus provides valuable information.

When looking at the detailed result obtained by the random positioning, Figure 4.16 and Figure 4.17, the presence of a laser inside an object does not necessarily penalise the final inference. On the 27 positions generated in total, 7 are in this case. This comprises the best performer, L_R_3, but not the worst, L_R_2. However, the presence of positions clearly visible from the scene corners is tied to a lower segmentation performance. The presence of position inside the scene is also apparent in well performing configurations. These observations gave the idea of visibility reduction.

4.2.4 Discussion

From these experiments on synthetic data topology, it appears that the three envisioned levels of details (layout, object, local point structure) are deeply interlinked. For example, the sampling process modifies the local point structure but provides different results following the scene layout and the mesh models used. A scene comprised of realistic point cloud objects is not useful when badly organised and repetitive. As such, it is difficult to precisely determine which topological aspect of a synthetic scene is more impactful on semantic segmentation performance. We can still consider scene layout as the primary candidate even if the scene is divided into chunk afterward. This is due to its influence on the other topological levels. After sampling, an object representation is modified due to occlusions occurring from its neighbours.

Experiments trying to further isolate those aspects can be devised but may be hard to carry out. To fully study the influence of local point structure, VSL could first be used to

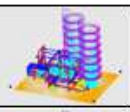
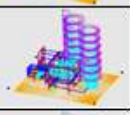
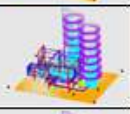
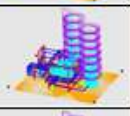
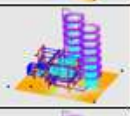
Training	Top view	Side View	Accuracy	mIoU	IoU Pipe	IoU Beam	IoU Valve	IoU Tank	IoU Floor	IoU Walkway	IoU Structural
L_A			72,94	22,50	72,21	27,33	4,11	38,64	81,79	51,38	36,83
*L_R_1**			72,90 ≈	22,72 ≈	69,87 -	30,04 +	3,88 ≈	37,92 -	82,95 +	**55,38** +	36,29 -
L_R_2			70,76 -	21,32 -	65,30 -	30,01 +	3,64 ≈	35,98 -	78,48 -	48,56 -	33,90 -
*L_R_3**			**74,35** +	**23,08** +	74,51 +	29,05 +	3,50 -	**41,89** +	82,63 +	49,21 -	**39,98** +
*L_R_4**			71,46 -	21,52 -	68,69 -	25,44 -	2,15 -	34,70 -	81,86 ≈	52,71 +	31,95 -
*L_R_5**			71,25 -	22,08 ≈	69,28 -	**31,40** +	**8,66** +	35,06 -	80,97 -	47,85 -	32,72 -
L_R_6			71,96 -	22,24 ≈	70,53 -	28,09 +	3,90 ≈	41,35 +	80,93 -	50,21 -	33,63 -
*L_R_7**			72,91 ≈	22,45 ≈	**75,28** +	28,12 +	2,88 -	41,02 +	**83,81** +	49,09 -	31,92 -
*L_R_8**			72,19 -	22,53 ≈	67,32 -	30,97 +	6,75 +	39,00 ≈	79,39 -	51,49 ≈	36,63 ≈
*L_R_9**			73,36 ≈	22,66 ≈	73,34 +	29,73 +	1,93 -	37,46 -	81,66 ≈	53,57 +	36,45 ≈
Occurrence when random is strictly superior to L_A (IoU R > A) (Blue)			2	4	3	8	2	4	4	4	1
+ : Superior occurrence (IoU R > A + 0.5)			1	1	3	8	2	3	3	3	1
≈ : Similar occurrence (IoU R = A ± 0.5)			3	6	0	0	3	1	2	1	2
- : Inferior occurrence (IoU R < A – 0.5)			5	2	6	1	4	5	4	5	6

Figure 4.16: Detailed comparison between random configurations and results obtained after using L_A. The * symbol denotes the presence of a laser position stuck in an object.

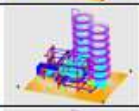
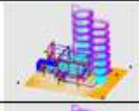
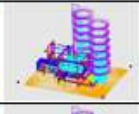
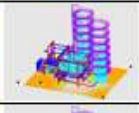
Training	Top view	Side View	Accuracy	mIoU	IoU Pipe	IoU Beam	IoU Valve	IoU Tank	IoU Floor	IoU Walkway	IoU Structural
L_AVR			74.33	**23.39**	73.22	29.90	2.42	**43.35**	82.93	53.02	39.89
*L_R_1**			72,90 -	22,72 -	69,87 -	30,04 ≈	3,88 +	37,92 -	82,95 ≈	**55,38** +	36,29 -
L_R_2			70,76 -	21,32 -	65,30 -	30,01 ≈	3,64 +	35,98 -	78,48 -	48,56 -	33,90 -
*L_R_3**			**74,35** ≈	23,08 ≈	74,51 +	29,05 -	3,50 +	41,89 -	82,63 ≈	49,21 -	**39,98** ≈
*L_R_4**			71,46 -	21,52 -	68,69 -	25,44 -	2,15 ≈	34,70 -	81,86 -	52,71 ≈	31,95 -
*L_R_5**			71,25 -	22,08 -	69,28 -	**31,40** +	**8,66** +	35,06 -	80,97 -	47,85 -	32,72 -
L_R_6			71,96 -	22,24 -	70,53 -	28,09 -	3,90 +	41,35 -	80,93 -	50,21 -	33,63 -
*L_R_7**			72,91 -	22,45 -	**75,28** +	28,12 -	2,88 ≈	41,02 -	**83,81** +	49,09 -	31,92 -
*L_R_8**			72,19 -	22,53 -	67,32 -	30,97 +	6,75 +	39,00 -	79,39 -	51,49 -	36,63 -
*L_R_9**			73,36 -	22,66 -	73,34 ≈	29,73 ≈	1,93 ≈	37,46 -	81,66 -	53,57 +	36,45 -
Occurrence when random is strictly superior to L_A (IoU R > A) (Blue)			1	0	3	4	7	0	2	2	1
+ : Superior occurrence (IoU R > A + 0.5)			0	0	2	2	6	0	1	2	0
≈ : Similar occurrence (IoU R = A ± 0.5)			1	1	1	3	3	0	2	1	1
- : Inferior occurrence (IoU R < A – 0.5)			8	2	6	4	0	9	6	6	8

Figure 4.17: Detailed comparison between random configurations and results obtained after using L_AVR.

create a scene where effects of its layout and objects geometry are visible (such as occlusions). The resulting point cloud could then be transformed back to a mesh containing occlusions information. By applying Random Surface Sampling to this second mesh, two scenes extremely similar in their topological aspect except for their local point structure could be obtained. Sadly, results of current state-of-the-art mesh reconstruction techniques are not precise enough. The test carried out resulted in meshes containing a high number of hole for objects far from the virtual lasers used. The surface of reconstructed meshes is also unnaturally rugged in most cases, which deform object models and thus beat the point of the process to isolate one topological level from the others.

Nonetheless, those experiments can be considered fruitful as they:

- Validate the benefits in using synthetic data to augment acquired data.
- Show our sampling method ability to improve the informativeness of synthetic data.
- Place the scene layout as the most important aspect when creating synthetic data.
- Show that focusing on domain shift exclusively is not the best strategy to use when class imbalance is high.

4.3 Synthetic data colouring

The devised colouring step is used to extract and transfer information from unlabelled point clouds to synthetic point clouds. This method is first tested on the S3DIS [9] dataset in Section 4.3.1. The result presented in this part were published in [23]. Tests on industrial data are then carried out in Section 4.3.2. Those tests prove to be less successful and reasons for this lack of success are investigated. Finally, some hypotheses behind the design of the colouring method are tested in Section 4.3.3 before a conclusion on the method can be made in Section 4.3.4.

4.3.1 S3DIS

To test the colouring method on S3DIS [9], the same dataset as experiments carried at Section 4.2.3 is used. This first test briefly checks the effect of the method but also the necessity of using a regularisation technique during its training. To do this, the synthetic data is coloured by networks trained on zone 2, 3, 4 and 6 of S3DIS with different combination of loss and regularisation.

Contrary to our hypothesis, compelling the colouring network to use more of the colour spectrum is counterproductive. As seen in Table 4.11, the colouring result is degraded. The obtained colour spectrums diverge from the original and the final segmentation performance decreases slightly.

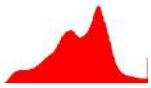
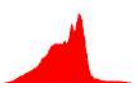
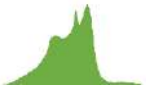
	Zone 1 only/Raw data	MSE	$MSE + R_{dist}$	$MSE + R_{var}$
Colouring				
Colouring				
Red colour spectrum				
Green colour spectrum				
Blue colour spectrum				
Inference with colour data (mIoU)	31.73	33.99	33.34	33.08
Inference without colour data (mIoU)	9.97	17.80	15.34	17.35

Table 4.11: Segmentation performance on the S3DIS zone 5 testing dataset depending on the colouring loss used to process training synthetic data.

The colouring network can be applied to acquired point cloud from zone 1 and 5 in order to see its performance. The MSE loss alone gives results relatively similar to the original: lights are coloured in white and doors have different colours than walls. The apparition of chromatic aberration is clearly visible in Table 4.11 when R_{var} is used. Whereas the use of R_{dist} creates a completely non-descriptive green-grey colouring. The blur present in the colouring obtained with the MSE loss is worsened by the addition of regularisation.

A problem of the colourisation technique becomes visible when looking at the colour spectrums: there is a peak in its middle that is absent from the ground truth. This effect is due to the MSE loss tendency to group data around a mean value. Both regularisations also worsen this problem. R_{var} partially succeeds in forcing the network to uses more of the colour spectrum at the cost of reducing its similarity with the original. The concern surrounding R_{dist} is verified as the network trained with the $MSE + R_{dist}$ loss games its score. This leads to a marginal uses of the spectrum extrema and a concentration of colours around the mean colour value.

Nonetheless, in each case, an increase in performance can be noticed when using synthetic data. Robustness towards loss of colour information is also improved. This first experiment invalidates the necessity to use a regularisation in the loss. The effectiveness of the colouring must still be proved. As such, the MSE loss will be used alone in a

Colouring	OA	mIoU	mIoU*	beam	board	book.	ceil.	chair	clut.	door	floor	table	wall	col.	sofa	wind.
Coloured	**71.1**	**34.4**	**43.4**	0.1	21.9	44.3	62.0	**65.0**	**37.5**	**35.5**	57.5	54.2	**70.6**	3.8	**9.5**	**19.0**
Colourless	69.1	33.1	41.7	0.2	**23.3**	**46.2**	59.7	64.9	36.9	28.7	54.1	56.5	67.9	4.6	8.0	13.0
Random colour	70.9	33.4	42.0	0.2	19.7	43.5	**63.1**	63.0	37.3	30.7	**60.6**	**58.2**	68.9	4.6	4.9	12.5
Zone 1 Only	67.6	31.7	39.9	0.1	18.7	44.2	60.7	61.5	37.1	27.7	46.4	53.6	68.0	**5.0**	7.7	13.7

Table 4.12: Differences obtained by changing the colouring of synthetic scene used in training.

second experiment.

This second experiment consists in an ablation study comparing segmentation results of network trained with data coloured by our method, randomly coloured data (each colour channel is drawn randomly following a uniform law) and colourless data (the colour channels are kept at maximum value: 255).

The use of colour increases segmentation performances (Tab. 4.12). The mIoU increases by 1.3 when the synthetic data are coloured by the network whereas randomly coloured synthetic data does not contain more information than colourless one. The tiny increase in performance when using them compared to colorless one (+0.3 mIoU) may be attributed to either a reduction in over-fitting risk or luck during the training. As such, it is not kept as a potentially useful data processing technique.

The score presented in Table 4.12 can be explained when looking at the result Figure 4.18. The degree of errors surrounding the floor and ceiling classes is reduced with coloured synthetic data. Some small geometrical details are also better detected, such as the board on the first line of Figure 4.18. The predicted semantic class of each point is also more coherent with its surrounding. For example, the bookcases in the fourth line of Figure 4.18 are better detected using coloured synthetic data: they are not cofounded with a wall and less of their points belong to another class despite the semantic classification of their neighbours.

Those results validate the usefulness of the colouring method when applied to the S3DIS dataset and dismiss the necessity to add a regularisation to the training loss.

4.3.2 SMARI

As shown in Section 3.2, industrial data composing the SMARI dataset and the S3DIS dataset are quite different. The applicability of the colouring method to data of industrial origin must be evaluated. In the following experiments, the same synthetic scenes as in Section 4.2.3.2, Synth3, will be used. The same algorithm, described in Section 4.1.3, colours all three.

Most experiments will focus on varying the data used for colouring. Four sets of data are considered and detailed in Table 4.13. Except for YAR, they represent unlabelled parts of industrial installations used to create the SMARI dataset.

Following the common philosophy in deep learning: “The more data the better” and

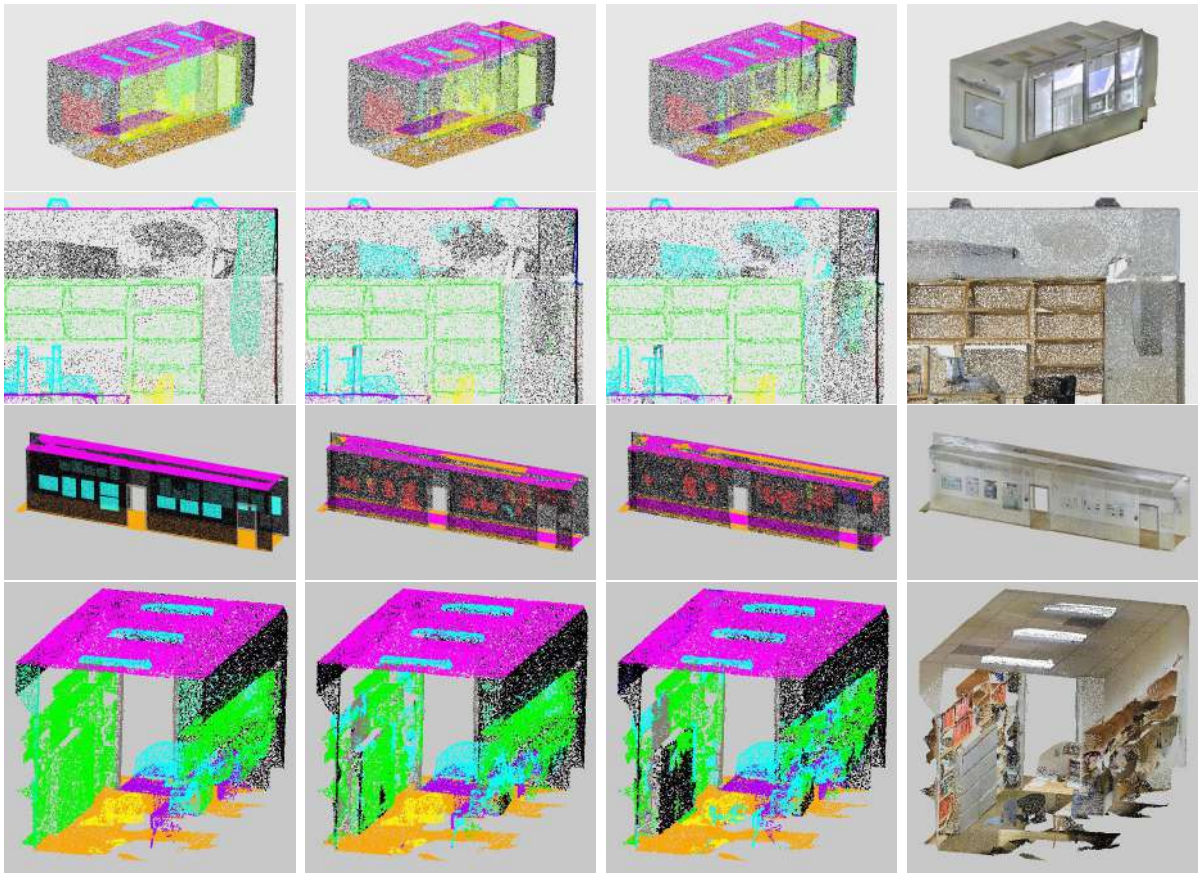


Figure 4.18: Semantic segmentation tests results obtained after training with synthetic data sampled with VLS, from left to right: truth, coloured synthetic data, colourless synthetic data and raw scene.

Name	Number of point	Surface (m²)	Characteristics
ADP	669 210 025	12 847	An indoor industrial facility represented with only a single colour channel. This channel is duplicated to imitate data represented with 3 colour channels.
DPO	375 440 730	23 844	A storage area represented with 3 colour channels and containing large tanks. Its colour pallet is mostly brownish.
KAM	355 577 643	21 082	A storage area represented with 3 colour channels which contains more piping and smaller tanks than DPO. It contains a ground full of verdant grass, metallic tanks and pipes painted with various colours.
YAR	343 183 465	4 777	An indoor industrial facility containing a high number of occluded areas. Lightning effects give it a brownish and blueish general colouring.

Table 4.13: Datasets used to train the colouring network.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired only	69.30	18.82	66.67	0.27	**5.76**	32.29	82.55	42.42	28.11
Colourless synthetic	73.17	22.65	**73.33**	27.54	3.98	40.69	82.60	**53.18**	33.10
Colour4	72.52	22.36	71.78	**30.47**	1.90	40.14	81.28	52.82	33.19
Colour3	**73.64**	**22.97**	72.61	29.02	5.54	**41.51**	**82.94**	50.27	**38.07**

Table 4.14: Inference results following the use of different colouring dataset.

because this worked quite well on S3DIS, the colouring network is trained on these four datasets at once (Colour4). To determine if the greyscale colouration of ADP has a positive or negative influence on colouration results, a second network is trained on DPO, KAM and YAR only (Colour3).

Contrary to the S3DIS dataset, the results obtained when synthetic data is coloured are not superior compared to colorless synthetic data (Tab. 4.14). Colour4 is slightly inferior to colourless data (-0.65 mIoU) and show a poorer performance on most classes, except beam. Colour3 being better than Colour4 in general (+1.12 mIoU), the influence of ADP is clearly negative. The results on beam obtained by Colour4 can be attributed to the high proportion of beam on ADP, which could have brought a more informative colouring on this class only. Colour3 is only slightly better than colourless (+0.47 mIoU) and does a better segmentation on most class. Walkways are less prevalent in Colour3 than in Colour4 and this class do not have a meaningful colouring in general, which could explain the lack of performance on it.

From this experiment, the influence of the colouring method in PCSS of industrial scene seems present but marginal or negative. As the method works on indoor data, the reason behind this lack of effectiveness must be explored. Does our industrial dataset possess characteristics absent from S3DIS which impede the colouring network? If so, can a more meaningful colouring be learned?

To investigate the first lead, a comparison of the colour distribution in both dataset is carried out and presented in Figure 4.19. From these histograms, it is quite clear that the datasets used in SMARI are dissimilar in their colour representation whereas the areas of S3DIS are homogeneously coloured. For example, YAR is darker than others and DPO colours are concentrated in the [25;100] range.

These differences could explain the lack of success encountered by the colouring method when applied to our dataset. To test this hypothesis, a colouring network is trained on each dataset separately. Segmentation results obtained after using each of those colouring network to synthetic data are reported in Table 4.15.

None of the four attains the performance of Colour3 or colourless data. The results obtained seem to indicate that using fewer data to train the colouring network also decreases segmentation quality. This could be explained by a difficulty to generalise the colouring to other scenes. Even so, in some cases, the segmentation network trained with

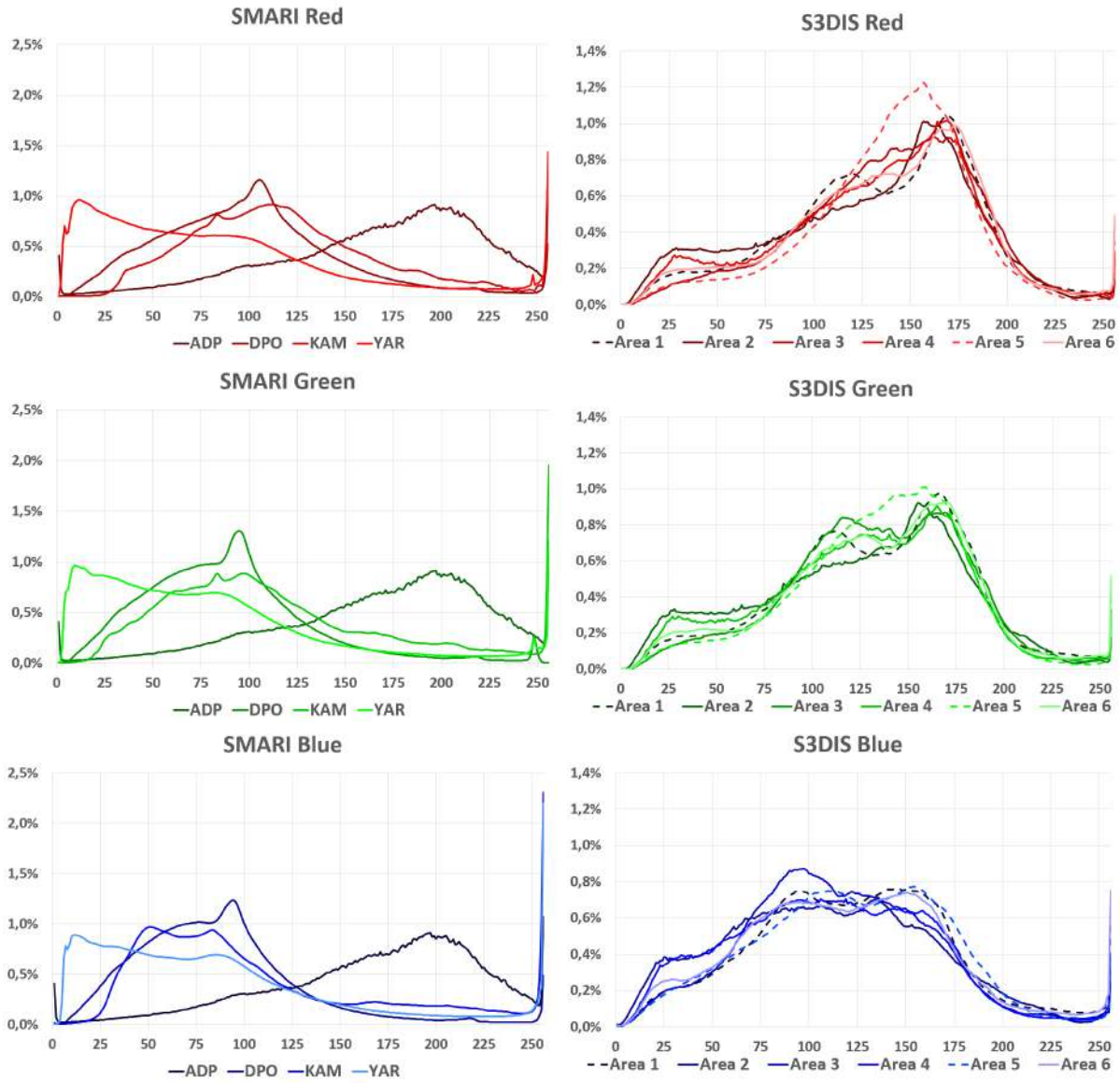


Figure 4.19: Colour spectrum of scenes used to train the colouring network, expressed in relative number of point per scene.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	69.30	18.82	66.67	0.27	**5.76**	32.29	82.55	42.42	28.11
Colourless synthetic	73.17	22.65	**73.33**	27.54	3.98	40.69	82.60	**53.18**	33.10
Colour4	72.52	22.36	71.78	30.47	1.90	40.14	81.28	52.82	33.19
Colour3	**73.64**	**22.97**	72.61	29.02	5.54	41.51	82.94	50.27	**38.07**
ADP	71.96	21.52	71.70	**31.16**	1.40	37.71	83.60	45.93	28.00
DPO	72.81	21.10	72.45	13.25	3.37	39.11	**84.37**	44.91	35.93
KAM	72.81	22.41	72.33	30.42	1.52	**42.08**	83.26	50.19	31.93
YAR	72.63	21.87	70.63	22.84	3.05	41.24	84.01	49.25	32.99

Table 4.15: Inference results following the use of different scenes to train the colouring network.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
DPO	72.81	21.10	72.45	13.25	3.37	39.11	84.37	44.91	**35.93**
DPO 2	72.22	20.86	70.94	12.13	2.33	39.07	84.04	45.95	34.64
DPO 3	71.81	20.63	69.82	12.87	2.34	37.55	84.70	46.02	33.94
KAM	72.81	**22.41**	72.33	**30.42**	1.52	42.08	83.26	**50.19**	31.93
KAM 2	72.36	21.54	**73.50**	24.67	1.41	36.82	84.07	47.54	30.58
YAR	72.63	21.87	70.63	22.84	3.05	41.24	84.01	49.25	32.99
YAR 2	**72.82**	21.87	72.73	24.13	**3.77**	**43.90**	**85.35**	44.45	30.12
YAR 3	71.92	20.89	72.90	15.43	1.53	39.33	84.12	46.92	30.29

Table 4.16: Repeatability of the colouring experiment.

the coloured data works better and reflect knowledge which seems to be extracted from the colouration network. This is the case with ADP for which segmentation of the beam class improves (+3.62 IoU compared to colourless) or KAM which yields better results on tanks (+1.39 IoU compared to colourless). However, the segmentation network can also be disturbed by the colouring, such is the case with DPO and the beam class, which falls by 14.29 IoU compared to colourless. This experiment shows that the colouring network does pass on information via colouring to synthetic data but also that this information is difficult to comprehend or uninteresting for the segmentation network.

Nonetheless, two additional knowledge can be learned with this experiment as a basis. The colouring information is well-defined by the colouring network and its extraction is relatively reliable.

By "well-defined colouring information" is meant that the colouring of each class is distinct from one another. This can be observed when looking at the colour histograms of some classes of synthetic objects once they are coloured (Fig. 4.20). Depending on the data used to train the network, the colours are also different. From those histograms, it is again possible to see the weakness of the colouring methods: the extrema of the spectrum are not used. However, even if the colours are concentrated on a narrower range than the original data, information is still conveyed to synthetic data. On one hand, some classes have a narrower profile which reflect the training data. For example, the floor is relatively uniform in both DPO and YAR. On the other hand, classes such as pipe has large colour profile due to their diversity in the training set, mostly because of painting or rust. This differentiation shows the ability of a network to learn the relation between form, colour and purposes of objects.

The extraction is relatively reliable as repeating the experiment produces similar results. When considering DPO, KAM or YAR, few changes were observed between experiments (Tab. 4.16). Some decrease in the understanding of the beam class can be observed sometimes in KAM and YAR whereas it remains constant for DPO. Except for this class, the results obtained from one experiment to another remain similar.

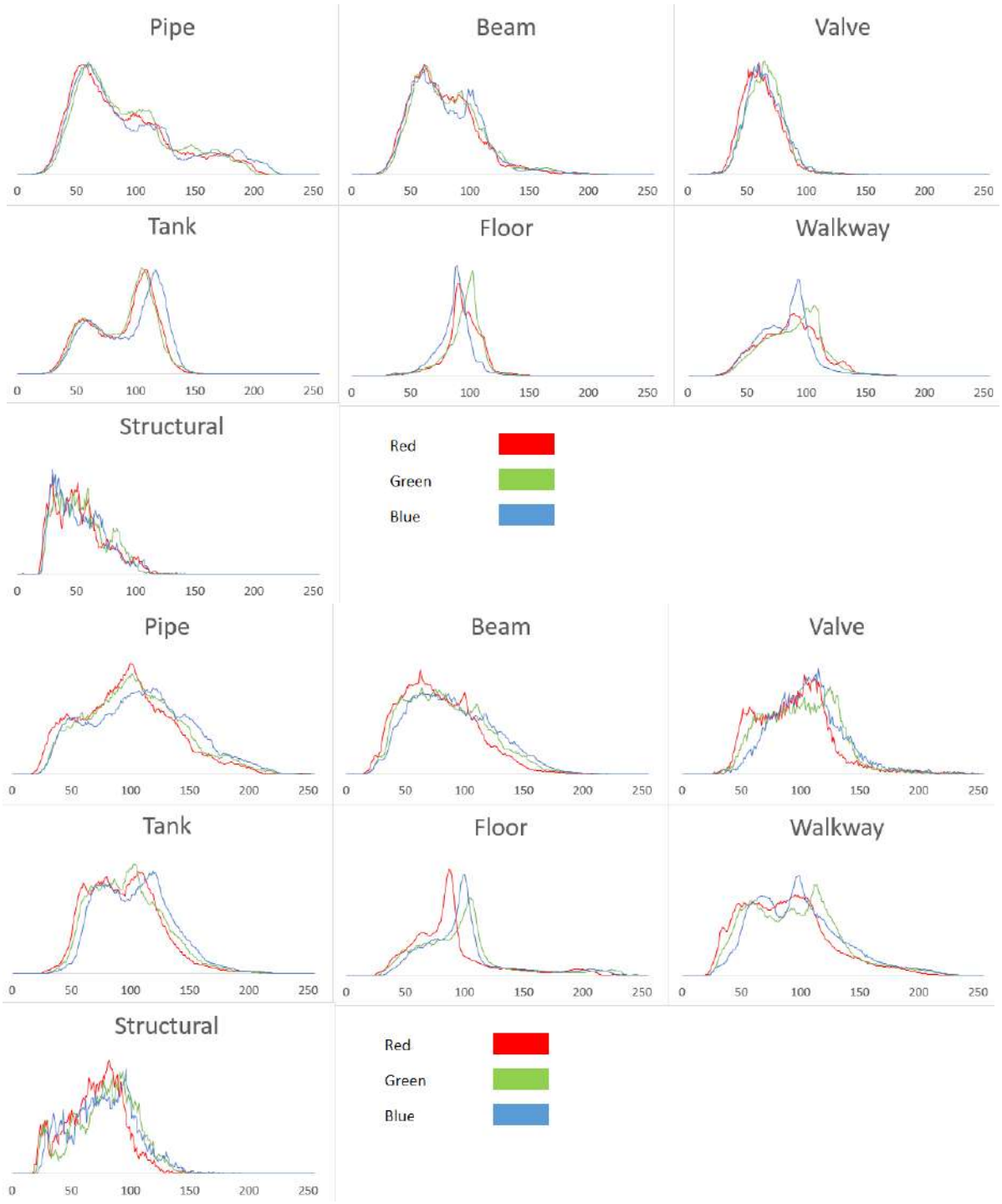


Figure 4.20: Colour distribution of synthetic objects once coloured by a network trained on DPO (top) or YAR (bottom)

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
DPO HSV	**72.81**	**21.10**	**72.45**	13.25	**3.37**	**39.11**	**84.37**	44.91	**35.93**
DPO RGB	71.32	20.52	70.92	**13.79**	1.97	39.01	83.32	**44.99**	30.71

Table 4.17: Segmentation results after changing the colour space used to train the colouring network.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
DPO	**72.81**	**21.10**	**72.45**	13.25	3.37	39.11	84.37	44.91	35.93
DPO Clipped $\mathcal{N}(0; 30)$	72.35	20.23	69.54	2.29	3.73	**39.20**	83.96	**46.41**	**36.11**
DPO Squeezed $\mathcal{N}(0; 30)$	71.22	20.64	68.22	**15.37**	**6.13**	36.17	**84.64**	45.88	30.36
DPO Clipped $\mathcal{N}(0; 60)$	70.05	19.29	68.77	3.56	3.24	36.06	83.56	46.23	26.22

Table 4.18: Segmentation performance after adding noise to the training data of the colouring network.

4.3.3 Hypotheses validation

Previous experiments showed working-cases and failing-cases of the colouring method. Two hypotheses surrounding this method must still be tested. First that the HSV colour space is a better fit to training a colouring method (H_{HSV}), secondly that the colouring method does learn the relationship between shapes and colours (H_{Rel}).

To test the first hypothesis H_{HSV}, the colouring network is trained on DPO using the RGB colour space. A general decrease in performance is observed (Tab. 4.17) when using the RGB space (-0.58 mIoU). The differences are mostly minor except for the pipe and structure classes. This is difficult to explain as there is no stark differences between the two when looking at the resulting colour distribution per class (Fig. 4.21).

A possible explanation would be our original intuition: some classes are sensitive to acquisition condition such as lightning and the use of HSV better reflect those cases. When looking at the same histogram in the HSV space in Figure 4.22, it is possible to see that the hue is more varied when the training occurs in the HSV space with some peak reflecting highly used hues in the training data. This is better than when training in the RGB space which only infers a reddish hue in most cases. H_{HSV} can be considered as validated.

To verify H_{Rel}, either the colour or shape of objects can be perturbed. The colour space was chosen and a random Gaussian noise is applied to the colours of DPO before training. Two methods are used to work around the cases where point colour goes outside the usual range of [0;255]. In the first case, the colour is simply clipped at the extrema. In the other case, the out of bound values are allowed and the whole spectrum is linearly squeezed in the [0;255] range after noise application. In both cases, the noise value applied at each point is the same before squeezing or clipping occurs.

As long as noise is used, the general segmentation performance decreases as reported in Table 4.18. Class by class, the result is more contrasted. The original DPO dataset

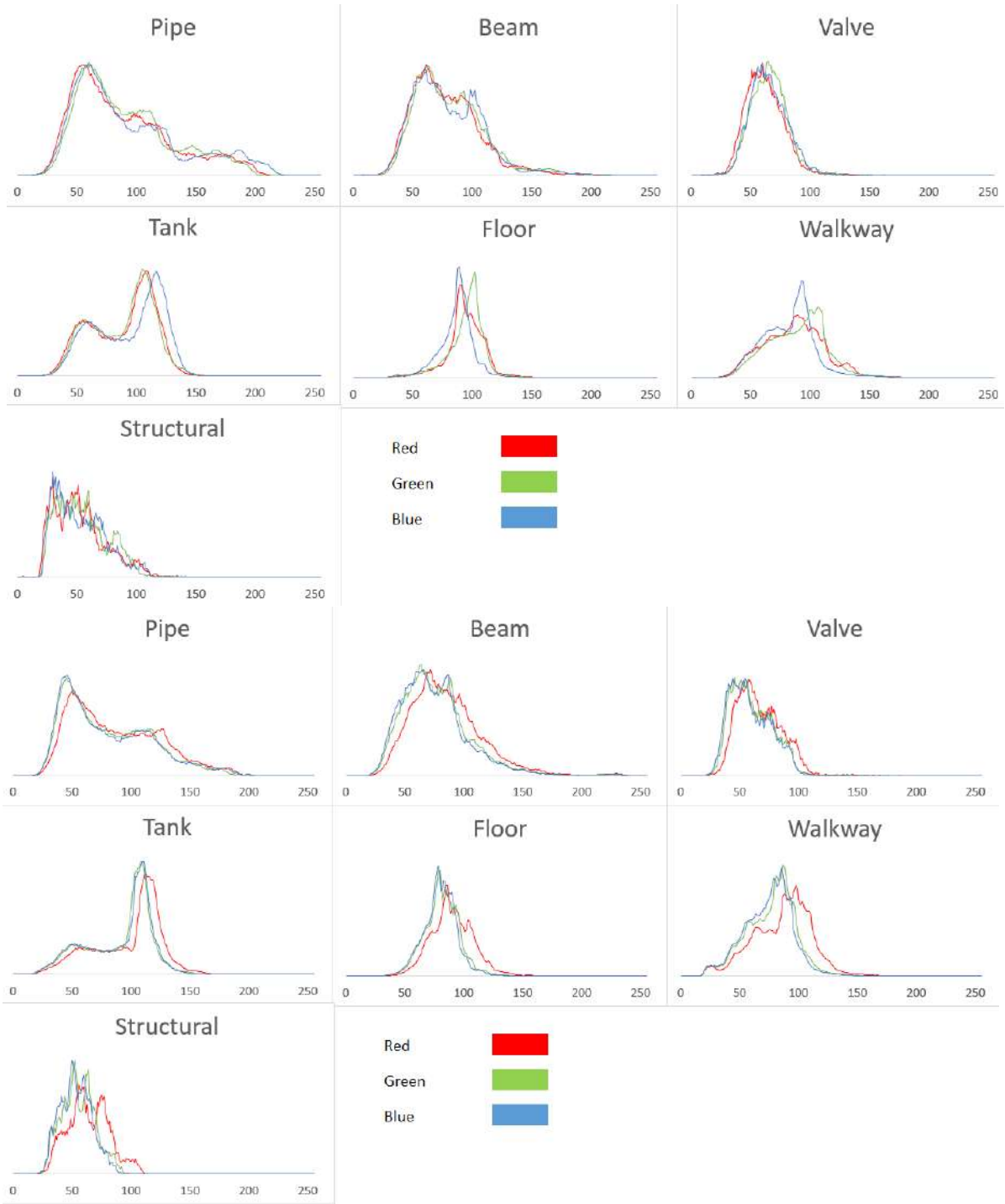


Figure 4.21: Colour distribution of synthetic objects once coloured by a network trained on DPO with the HSV colour space (top) or RGB colour space (bottom)

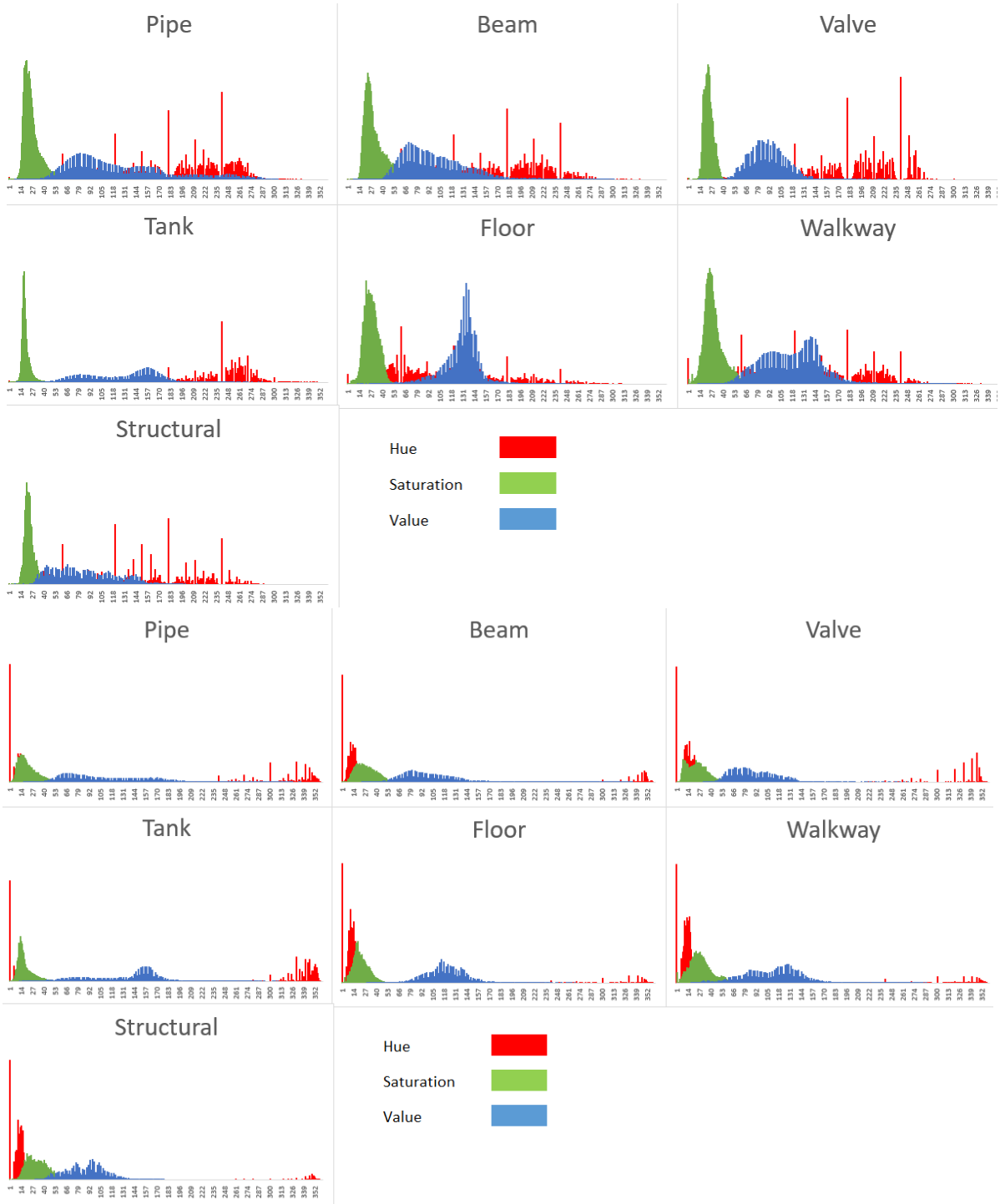


Figure 4.22: Colour distribution of synthetic objects once coloured by a network trained on DPO with the HSV colour space (top) or RGB colour space (bottom)

performs relatively well in each case but is only better at segmenting the piping class. For other classes, the $\mathcal{N}(0; 30)$ noise performs slightly better, with a minor advantage when squeezing is used on border cases. Most notably, clipping removes the network ability to understand the beam class (-10.96 IoU). Moreover, a stronger noise ($\mathcal{N}(0; 60)$) further decreases the general segmentation performance. H_{Rel} is thus verified.

4.3.4 Discussion

The current colouring method is a partial success. It adds information to synthetic data when the domain of application is coloured homogeneously. However, it fails to add enough meaningful information when the domain is coloured heterogeneously. Thus, for industrial data, it does more harm than good. The last experiments show ways to improve this method by modifying its data: either by using a more homogeneous training dataset or by adding a slight noise to training data.

Experiments to train the colouring network by a generative adversarial process were also attempted. However, using a classic GAN structure failed as training lead to mode collapse. Using a Wasserstein GAN also failed due to the gradient penalty growing uncontrollably. Efforts to apply this technique to improve colouration performance are thus ongoing and planned as further work after this thesis.

4.4 Conclusion

In this chapter, a synthetic data generation process was explored. This process contains two new methods. One to transform synthetic mesh scene in point clouds, the other applies colour to the generated colourless synthetic point cloud by extracting information from unlabelled acquired data.

Experiments showed the effectiveness of adding synthetic data to an acquired dataset and validated the performance of the sampling method. Computing good virtual laser positions is an effective technique when working on a scene comprehension task. The use of a full-wave simulation process generates a good noise model that seems sufficient to bring out the potential of synthetic data. Adding a Point Cloud Translation model [103] could potentially further improve this technique.

The results on the colouring process are more nuanced as the method only works on datasets where the colouring is homogeneous and regular. Thus, this method works on a dataset from the literature which is sampled from a single location but fails in the case of our industrial data which come from multiple locations.

The current method focuses on using existing 3D mesh models, such as CAD models or ones made from 3D modelling software. An automatic synthetic scene generation process is only entertained in the case of interior environment or briefly done for the sake of one

experiment. Creating such an automatic scene generation process could be a consequent improvement to the current synthetic data generation process.

Chapter 5

Data representation

Different concepts linked to point cloud data in general are studied in this chapter. Those studies allow answering RQ2: the way data is presented to the semantic segmentation network does influence its performance. First, strategies to divide point clouds in data chunks whose size is manageable by the semantic segmentation network are explored. A strategy computing pillar shaped chunks overlapping with each other is selected. Following this exploration of data division, a strategy to improve the colouring method is devised. This strategy use several colouring networks, each trained on data acquired from the same location, to create multiple versions of the same synthetic data coloured differently. A ratio parameter is defined to control the relative quantity of acquired and synthetic data seen by the segmentation network during training. Using a ratio of 0.5, segmentation mIoU increases by 0.51 compared to the best previous colouring method. In a second part, data augmentations are studied. Computing point normal prove more successful as a point augmentation technique than computing point curvature. A random rotation around the vertical axis applied to each data chunk during training also proves effective to make the network robust towards rotation of data. Finally, the classification of data chosen initially is questioned, a simpler and more effective classification is determined. It groups objects belonging to the same system, such as piping or electrical, in a single class. This allows the network to focus less on small differences between highly represented classes with similar functions, such as structural and beams, and focus more on the less represented classes with completely different uses in the industrial domain, such as electrical components.

Studying the literature showed that using different feature extraction kernels and neural network structures influences how information is extracted from data during training (Sec. 2.3). In Chapter 4, ways to add synthetic information to acquired data were studied. Nonetheless, the possibility to extract more knowledge from data by modifying them simply before they are seen by the segmentation network during training remains to be studied (RQ2). Two things can be controlled and modified in the input data: shape and

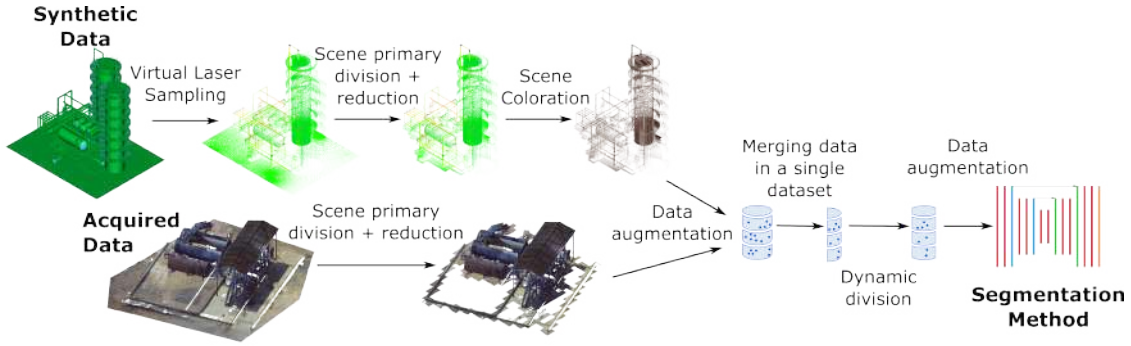


Figure 5.1: General data processing workflow proposed in this thesis.

content. Modifying the input shape is studying how different division strategies of large point cloud scenes to data chunks manageable by the network influence segmentation performance (Sec. 5.1). The content is the information present in each chunk, and the act of modifying this content to help the network during training is known as data augmentation (Sec. 5.2). In both cases, the modification can be either static or dynamic.

Static modifications are pre-computed before the training process is launched and present the advantages to be a one-time-cost. It is the preferred approach for modifications that are computationally expensive. The colouring method presented Section 4.1.3 can be though as a static augmentation method applied to synthetic data.

Dynamic modifications are applied to data during the training process each time a new batch of data is necessary to teach the network. They are preferred when the modification is less computationally costly. Their main advantage is the diversity that they bring without occupying more disk space. Pre-computing the results of some modifications methods can multiply the space necessary to save the training dataset while bringing less diversity to what the network see during training.

When looking at our general data processing workflow as presented in Section 3.3.2 (Fig. 5.1), modifications can be applied once a point cloud is obtained, either by real data acquisition or after Virtual Laser Sampling for synthetic data.

A last influence on segmentation quality is the labelling applied to data. As seen in Chapter 3, our data classification was made after studying the data available and consulting industrial draughtsmen. However, several flaws are apparent in this classification, notably the impossibility to segment seven out of the fourteen classes. Having half of the classification thought of as meaningless by the segmentation network is a problem, even when we knew beforehand that some classes are far less important and represented than others (Sec. 3.1). Attempts to rectify the classification using the knowledge obtained during this thesis are presented Section 5.3.

Following those works, a conclusion around our second research question is made in Section 5.4.

5.1 Scene division

Point cloud scenes and point cloud semantic segmentation networks are more resource intensive than their 2D counterpart. As such, most methods in the literature do not work on the whole scene. The classical way to proceed is to divide the scene in chunks that are small enough to be manipulable by the segmentation network. Most methods follow the initial methodology of PointNet [75] and use vertical pillars (Sec. 2.3.3.3). As industrial scenes are peculiar in their structure (Sec. 3.2.1), the validity of the current scene dividing methods must be checked. Several division methods are first presented in Section 5.1.1 before experiments are carried out on static division (Sec. 5.1.2) and dynamic division (Sec. 5.1.3). Some strategies used during these experiments are then applied in Section 5.1.4 to enhance the colouring method devised in Section 4.1.3.

5.1.1 Division strategies

Two division processes are devised, one static and one dynamic. The static one computes chunks based on whole scene data similarly to the approach used before when dividing the scene in pillars with a square base of 1 by 1 meter. The dynamic approach computes chunks each time the network need data during training.

The process used for static division is described in Algorithm 4. For this method, the chunk shape influences the size parameter definition, the bin process but also how the step parameter is applied. When working with cubic shapes, extrema of the to-be-divided scene are taken to initiate the grid of bins instead of a grid following fixed coordinates (such as integer). This reduces the number of chunks containing little information in most cases. For circular shapes, the first bin centre is initialised from the extrema. The step controls the distance between the origin of each bin and is a measure of overlap. Due to the large size of the data used, the implementation of this algorithm works on data stream. This design choice allows to use the same process to divide extremely large point clouds representing whole industrial installations to smaller scenes.

Algorithm 4 Point cloud static division algorithm

Require: Point Cloud P, Chunk shape $ChunkShape$, Step $Step$
Ensure: $P.Length > 0$, $ChunkShape.Size > 0$, $Step.Length \geq 0$
  $SceneSize \leftarrow P.ComputeSize()$
  $Bins \leftarrow ComputeBins(ChunkShape, Step, SceneSize)$
  for all p **in** P **do**
    $Bins.Bin(p)$
  end for

After the division process, the reduction step, as described in Section 3.3.2, is applied to remove data chunks of little informativeness.

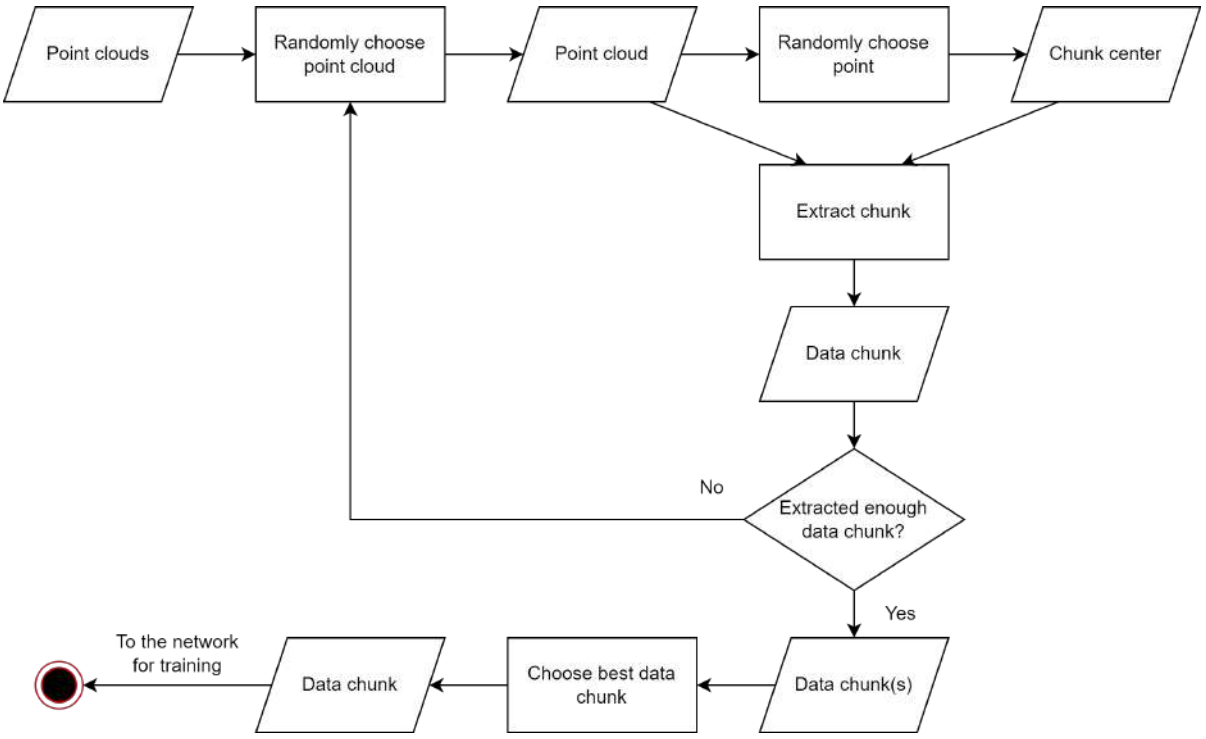


Figure 5.2: Schematic representation of the dynamic division process.

In the case of dynamic division, it is applied directly on data fed to the network and follows the process illustrated in Figure 5.2. The shape of the initial point clouds depends on the size of the scenes in the original dataset. A static division process can be applied beforehand to reduce the computational cost of dynamic division. This is the case with the acquired SMARI data. When a natural division of the data is available, such as rooms for the S3DIS dataset, those are used as initial point clouds. Contrary to the static process, the dynamic division goal is to generate one chunk of data to be fed to the network. As such, the point defining the chunk is drawn randomly from the available data and considered as its centre for extraction. Once again, the exact extraction method depends on the desired size and shape of the chunk.

To feed one data chunk to the network, more than one chunk can be extracted from the available data. Three different strategies are considered:

- A baseline random strategy, which extracts only one data chunk.
- A number strategy, which chooses one chunk from several depending on the number of points per class in each chunk.
- A frequency strategy, similar to the number strategy but using class frequency instead of number of points.

The first strategy is the least costly but does not offer control on the data given to the network. The last two try to dynamically balance the data. In [39], Griffiths and Boehm

showed that increasing the quantity of data chunk containing underrepresented classes enhanced semantic segmentation performance on the ScanNet [30] and Semantic3D [41] datasets. The number and frequency strategies try to reproduce this result dynamically.

During the whole training, the division algorithm keeps track of the class allocation in data given to the network for training. When the algorithm needs to choose between different data chunks, it compares the variance of the class distribution obtained by adding each data chunk and chooses the one minimising this variance. As the SMARI dataset is greatly unbalanced, this strategy will favour the rarer classes as long as they are present in one of the randomly created chunks. The class favoured may depend on the number of created chunks. When only a few chunks are generated, this strategy favours classes whose objects are often represented in data, even if few points describe them. If more chunks are used, classes with few instances will have more chance to be drawn and thus will be favoured. Exceptions could be large objects such as tanks as they are relatively large and represented by an important number of points. The chance of one being present in a chunk is high compared to the total number of instances in the dataset. A danger in considering numerous chunks at once for this strategy is the possibility to quickly overfit or focus too much on one class. If objects of one class are rare, they will have a higher chance to be selected if more chunks are considered. However, as they are rare, few variations in data will be offered to the network if the only data selected are a few variations of a handful of scene chunks. Thus, by design, this method cannot create a "balancing miracle" for classes present in only a few chunks but can work with moderately rare classes, such as valves in our case.

Those two strategies used to choose data chunks can be expressed more formally. Let $|C|$ be the cardinality of the semantic classes set as defined in Definition 4. Let A be the general notation for vectors of size $|C|$ representing data allocation following semantic classes. In those vectors a_i represents the current allocation surrounding the ith class. The allocation vector representing the allocation of the data already seen by the network during training is A_{seen}. The allocation vectors A_i represent the data distribution of the ith generated chunk. For $|A|$ newly generated chunk, both strategies can be described as:

$$\min_{i\in[\,1;|A|\,]}(\sigma^2(A_{seen}+A_i)) = \min_{i\in[\,1;|A|\,]}(\frac{1}{|C|}\sum_{j=1}^{|C|}(a_{seen\_j}+a_{ij}-\overline{A_{seen}+A_i})^2) \tag{5.1}$$

The difference between the two strategies lies in the way of computing A_i for each chunk. When considering a point cloud P as in Definition 1, a class representation can be considered as an application on this point cloud ($A\colon \mathbb{R}^3 \to \mathbb{R}, P \mapsto A(P)$). In this case, A_{Number} can be defined as:

$$A_{Number}(P) = \begin{pmatrix} \sum_{j=1}^{N} l_1(p_j) \\ \sum_{j=1}^{N} l_2(p_j) \\ \vdots \\ \sum_{j=1}^{N} l_C(p_j) \end{pmatrix} \; with \; l_i(p_j) = \begin{cases} 1 & if \quad c_j = c_i \\ 0 & otherwise \end{cases} \tag{5.2}$$

And with N the size of the considered chunk, $A_{Frequence}$ differs from A_{Number} by a single coefficient:

$$A_{Frequence} = A_{Number}/N \tag{5.3}$$

However, even if they differ by a single coefficient, the two strategies consider different information. The number strategy considers the training process as a whole by using raw numbers and tries to balance the overall number of points of each class seen by the network during training. The frequency strategy is more focused on data allocation in each chunk. Moreover, each chunk has the same intrinsic weigh when using the frequency metric whereas the gain brought by each new chunk diminish quickly with the number strategy in front of A_{seen}.

Following their definition, the static (Sec. 5.1.2) and dynamic (Sec. 5.1.3) division processes must now be tested.

5.1.2 Overlap and shape

This first series of experiment focus on the static division process. It has three goals. First, evaluating the necessity to overlap each data chunk. Second, to see if different shapes of chunk can have a positive influence on segmentation performance. Third, looking into a possible positive trade-off between batch size and the number of point considered per data chunk. Until now, the step parameter in the division algorithm was identical to the size of the shape used. Overlapping ($step < size$) data chunk would generate more data but allows the network to see the same objects in different context during training.

5.1.2.1 Overlap

In this first experiment, two strategies to create overlapping data are studied. The first strategy, O_{st}, is straightforward, takes the original scenes and divides those scenes while overlapping each data chunk. The second strategy, O_{rc}, recombines already divided and reduced scenes before dividing them again. In both case, the generated data is still represented as vertical pillars with a 1 by 1 meter base and a 0.5 meter overlap ($size - step$) in both direction is used. Using the recombination strategy O_{rc} should nonetheless generate pillars with a higher density of useful information. The acquired SMARI dataset and Synth3 are both used for this experiment. As we previously saw in Section 4.3,

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	69.30	18.82	66.67	0.27	5.76	32.29	82.55	42.42	28.11
No overlap	73.17	22.65	73.33	27.54	3.98	40.69	82.60	53.18	33.10
O_{st}	**77.02**	25.06	**76.53**	**32.81**	7.33	43.91	**84.63**	57.86	45.62
O_{rc}	76.36	**25.27**	74.57	30.32	**12.68**	**46.56**	80.89	**60.06**	**45.80**

Table 5.1: Inference results following the use of overlapping data to train the network

knowledge on the colour information is still shaky. As such, colourless data is considered in the following experiments. The straightforward and recombination strategy creates 7124 and 8680 pillars respectively from the 1829 original pillars.

The results obtained after training show a substantial increase in segmentation performance (Tab. 5.1). In both case, training with overlapping data allows the network to perform better on general and on every class. The only exception being the floor class in the recombination case. However, this is not problematic as this class is not a priority target and is sufficiently well segmented. Between the two strategies, the only class where the difference is large is the valve class (+ 5.35 IoU) where O_{rc} performs better. This should be due to the superior information density when using this strategy. However, recombination does not offer substantially better general performance as the created data pillars also contains zones of emptiness. As this trade-off only benefits one class, both methods can be considered pretty similar performance-wise.

The performance increase from such method is noticeable but costly. Multiplying the training dataset size by 4 multiply the training time by the same coefficient. Thus, the informativeness of the dataset as a whole augment, but in a very brute-force manner. This confirms that overlapping data can be beneficial but also justify further experiments on geometrical shape and size of data chunks.

5.1.2.2 Shape

Two kinds of geometric shapes are considered: pillars and balls. For the pillars, several sizes of square and cylindrical bases are considered. A ball shape is defined by its associated distance metric. Only the two most common are considered, L_2 and L_{max}, creating spheres and cubes. Circular and square shapes are considered in both cases due to the nature of the segmentation network. The KPConv kernel, when not deformable, is contained in a ball [92]. Having data chunk of similar shape could be potentially beneficial to segmentation performance. The chunks are computed before training using the straightforward strategy. Each combination of shape, size and overlap is named using the following convention: $Shape_{size\_overlap}$.

The networks trained on those data are compared using the classical testing dataset of shape SP_{1_0}. The characteristics associated to those new forms of our dataset are detailed in Table 5.2. The training time is expressed as a factor relative to the time needed to

Name	Shape	Size	Overlap	Nb of chunk	Nb of point	Ratio	Train time
SP_{1_0}	Square pillar	1×1 m	-	1829	69 420k	38 138	1.0
SP_{1_05}	Square pillar	1×1 m	1.0 m	7124	270 368k	37 951	4.0
SP_{2_1}	Square pillar	2×2 m	1.0 m	2450	289 563k	118 189	1.8
SP_{2_05}	Square pillar	2×2 m	0.5 m	1077	131 464k	122 065	0.8
CP_{1_1}	Cylinder pillar	1.0 m radius	1.0 m	2317	237 559k	102 529	1.6
CP_{15_15}	Cylinder pillar	1.5 m radius	1.5 m	2764	571 166k	206 645	3.3
CB_{2_05}	Cube	2×2×2 m	0.5 m	4321	153 965k	35 632	2.2
SB_{15_05}	Sphere	1.5 m radius	0.5 m	19762	1 076 150k	54 456	10.7

Table 5.2: Data chunks used in the experiment on shape and overlap.

train the network on the 1 m × 1 m square-based pillar data chunk on the same machine.

Results obtained on the testing set of usual shape (SP_{1_0}) are detailed in Table 5.3. A first view of these results show that SP_{1_05} is still the best performer. In fact, no new combination of data shape and overlap has a higher mIoU and accuracy than the initial SP_{1_0}. When linking those results with the chunks shapes, the number of point, the number of chunk or the ratio between those two, a few things can be concluded.

First, having chunks whose shape cuts their connection to the ground reduces the segmentation ability of the network. The problem is somewhat mitigated with SB_{1_05} as its large radius captures a large part of the scene. However, considering its tremendous training time[1], this is unsatisfactory. For industrial application, ball-shaped chunk must be avoided. This contrast with findings on the segmentation method made by its original author where ball-shaped chunks performed well on interior and urban datasets [91]. This difference is certainly due to the industrial domain characteristics as explored in Section 3.2. Experiments made on S3DIS [10] in Section 5.2.2.2 challenge the findings of Thomas et al. [91] when the division strategy is used in isolation.

Having a circular or square base does not seem to have an observable influence on segmentation performance. This is true when considering both pillar and ball chunks, and thus refutes our ideas behind modifying the shape. However, having a higher ratio of points per chunks seems to negate the advantages of the overlap. SP_{2_1}, SP_{2_05}, CP_{1_1} and CP_{15_15} all have more than 100k points per chunk. Only 4096 points are considered by chunk by the network. Those points are sampled with furthest point sampling and thus represent well the whole chunk. But they still contain less than 4.1% of the chunk information. Even if they cover the whole chunk, they cannot represent detailed information. When considering the intricacy of the industrial setting, large chunks do not seem ideal. A minimum density of point in the chunks could be necessary to attains good performances.

1. Only one network was trained with this division of data. This decision was made after considering the lacking performance offered by the network and the training time: half a week.

Data shape	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
SP_{1_0}	73.17	22.65	73.33	27.54	3.98	40.69	82.60	53.18	33.10
SP_{1_05}	**77.02**	**25.06**	**76.53**	32.81	7.33	**43.91**	**84.63**	**57.86**	**45.62**
SP_{2_1}	69.24	20.61	67.51	**35.19**	0.70	33.88	79.04	51.00	19.35
SP_{2_05}	69.59	21.84	61.95	34.61	7.63	35.99	79.89	53.15	30.49
CP_{1_1}	72.61	22.55	66.91	27.68	3.66	41.27	79.64	55.55	38.36
CP_{15_15}	71.10	21.79	70.74	32.46	5.68	38.10	80.89	49.58	26.21
CB_{2_05}	67.12	19.60	69.47	33.23	2.19	25.73	79.24	44.69	17.78
SB_{1_05}	69.21	21.12	67.99	27.57	**11.02**	31.96	80.36	48.66	27.10

Table 5.3: Inference results following the use of different shape of data chunk to train the network.

Data shape	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
SP_{2_1} 4096	69.24	20.61	67.51	**35.19**	0.70	33.88	79.04	51.00	19.35
SP_{2_05} 4096	69.59	**21.84**	61.95	34.61	**7.63**	**35.99**	79.89	53.15	**30.49**
SP_{2_1} 8192	70.54	21.21	67.97	33.29	2.06	34.45	79.47	50.94	26.93
SP_{2_05} 8192	**71.48**	21.72	**70.49**	33.90	2.46	31.67	**82.01**	**53.75**	28.34

Table 5.4: Inference results following an increase of points per chunk and a decrease of batch size.

5.1.2.3 Number of points

This hypothesis can be checked by training the network with more point by chunk. A new trade-off is explored by using 8192 points per chunks with SP_{2_1} and SP_{2_05} but dividing by two the batch size (16 to 8).

This trade-off does not bring great changes to the network capabilities (Tab. 5.4). The accuracy increases for both SP_{2_1} and SP_{2_05} by 1.30 and 1.89 respectively. However, the mIoU only increases a bit for SP_{2_1} (+0.6) and decreases slightly for SP_{2_05} (-0.12). The performance class by class also becomes similar between SP_{2_1} and SP_{2_05}. When only 4096 points are considered, SP_{2_05} segments objects of the valve (+6.93 IoU) and structural (+11.14 IoU) classes better than SP_{2_1}, for a decrease in pipe segmentation capability (-5.56 IoU). With 8192 points per chunks, the biggest difference is on the walkway class (+2.81 IoU). Moreover, SP_{2_1} catches on the structural class and SP_{2_05} on the pipe class.

From these first experiments on data division, it seems that bringing more diversity in data by overlapping chunk is beneficial to the segmentation network training. However, a static division process generate a high amount of information which greatly increases training time (x4). Trying to modify the shape of the chunk does not bring significant performance difference and is thus not further pursued. This nonetheless allowed us to verify the impact of one of the specificity of industrial data: their verticality. Having data also divided along the vertical axis is not beneficial to segmentation performance. Finally, a minimum density of point per chunk must be observed to maintain quality

Data shape	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
SP_{1_0}	73.17	22.65	73.33	27.54	3.98	40.69	82.60	53.18	33.10
SP_{1_05}	**77.02**	**25.06**	**76.53**	**32.81**	7.33	43.91	**84.63**	**57.86**	**45.62**
Random	71.45	21.25	74.38	29.50	2.66	32.38	81.41	51.71	23.88
Frequency	63.76	19.43	55.55	29.07	**12.96**	32.36	71.49	43.03	23.31
Number	60.85	18.23	49.59	23.91	0.45	**46.70**	71.22	35.68	27.19

Table 5.5: Inference results following the use of different dynamic division method applied before the network input during training.

results. Trading batch size for an increase of point per chunk is not beneficial. However, this is mostly a material limitation and this lead could be explored again when machine with more VRAM will be available. For now, a dynamic division of the scene seems to be the most pertinent way to move forward.

5.1.3 Dynamic division

Following the previous observations, the dynamic sampling methodologies devised in Section 5.1.1 are tested. As with Section 5.1.2, the acquired SMARI dataset and Synth3 are both used for this experiment. The dynamic divisions are initially carried on the pre-computed and reduced SP_{2_1} division.

5.1.3.1 Initial experiment

Following the observation of the obtained results (Tab. 5.5), it is apparent that there is a problem with the dynamic division method. The mIoU of each method is inferior to training with static data, with or without overlapping. This decrease in general performance is followed by a decrease of segmentation capability for each class in the case of the random process. Theoretically, the random division process should have results comparable to at least SP_{1_0} and should tend to approach SP_{1_05}. A difference in the data obtained by those two processes could explain this discrepancy.

Nonetheless, a few positive conclusions can be gathered from this experiment. First the influence of the floor class is clearly reduced when using the Frequency and Number drawing strategies (-9.92 and -10.19 IoU respectively). The improvement concerning the valve class brought by the frequency strategy (+5.63 IoU) is encouraging. After improving the general dynamic division process, this strategy could also be improved. Finally, the number strategy performances are underwhelming and can be put aside. It only improves performance on the tank class (+2.79 IoU), a class which is already highly present in the training data. From this experiment, it appears that a dynamic balancing division method should focus on point frequency and not number as a metric.

Looking at an extract of the cumulative points allocation generated by each strategy further confirm the number method inadequacy. To visualise what is presented to the

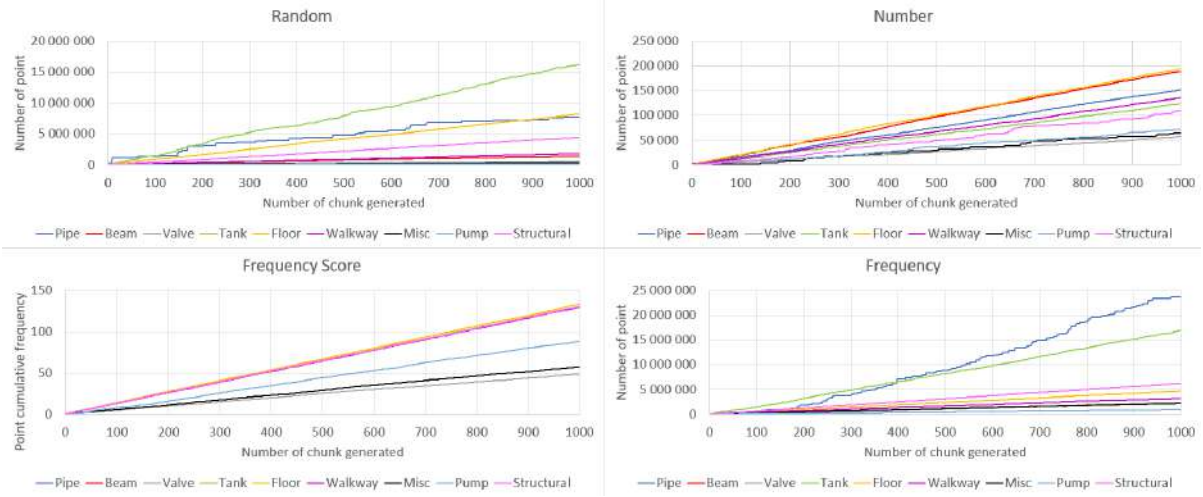


Figure 5.3: Number of point generated per class for the different strategy.

network each time a new chunk is extracted from SP_{2_1}, a thousand chunk is generated following each strategy. This process is repeated eleven times and the mean evolution of the number of points per class is represented in Figure 5.3. The cumulative point frequency is also presented in the frequency strategy case to evaluate what information is used to guide each strategy decisions.

Quite clearly, each strategy presents different information to the network. The random strategy roughly follows the class distribution of the dataset after reduction. The smoothness of each line depends on the probability of each class to be present in a chunk. The frequency strategy focuses much more on the pipe class than others. Strangely, this dedication to this class is not felt in the final segmentation results. It also succeeds in bringing rarer classes to attention than the random strategy, such as valves, miscellaneous or pump. Interestingly, cumulative frequencies of the most represented classes in the dataset are extremely similar. The strategy partially attains its goal. The number strategy gives a more balanced data allocation to the network but fail catastrophically in supplying enough point to the network. Compared to the other strategies, it is off by two orders of magnitude. This could be the principal reason of the method failure. The reason behind this lack in number of generated point must be investigated. Is the failure case of focusing too much on similar sample imagined in Section 5.1.1 be verified?

To answer this question, a thousand chunk of data are generated with each method. An approximate surface is computed using CloudCompare by grouping the points in square cells of 1 cm. When looking at those surfaces, represented in Figure 5.4, it is clear that the number method fail in presenting the whole dataset to the network during training. The frequency method is less faulty but concentrate on some areas a bit too much nonetheless. The surface values corroborate those observations. For 1000 chunks, 347, 202 and 50 m^2 of surface are generated by the random, frequency and number strategies respectively.

Following the failure of these two strategies, several possible improvements are ex-

Figure 5.4: Approximate surface covered by 1000 chunks generated with each dynamic method. The method from left to right is: random, frequency, number.

plored. First, the size of the chunks used as a basis for the dynamic division is increased. Then the possibility to perturb the frequency strategy in order to increase its performance is investigated. Finally, a compromise is searched between SP_{1_05} and the random strategy by randomly drawing pre-divided SP_{1_05} chunks.

5.1.3.2 Initial chunk size

Before focusing on the balancing strategy, the general way dynamic division is performed should be studied. One hypothesis around the lacklustre results is an increased probability in creating information poor chunks. Those information poor chunks are created when the dynamic division takes a too small part from the pre-computed chunk. This can happen when the centre point is selected at the edge of the pre-divided data. As such, only a small surface area of the scene is used to train the network. Those small chunks have more risks to contain few points or only one semantic class.

When considering the simplest case, a 2 m^2 surface with a uniform point distribution, there is a 75% risk of creating a chunk with a surface inferior to 1 m^2. The acquired data mean surface after pre-division is 3.762 m^2 and the mean surface of SP_{1_0} is 0.947 m^2. With a uniform point distribution, there is a 77% and a 75% chance to create a chunk whose surface is smaller than 1 m^2 and 0.947 m^2 respectively. This can be thought as a serious downgrade in data quality. To improve data informativeness when dynamic division is used, the sizes of the pre-computed chunks are multiplied by 2, which give them a theoretical surface of 16 m^2 and a 43.8% probability to create a chunk smaller than 1 m^2.

Following this idea, the data is divided into chunks of shape SP_{4_1} and the Random dynamic division strategy is used to train the segmentation networks. However, contrary to our hypothesis, the performance of the segmentation network decreases when using

Data shape	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
SP_{1_0}	73.17	22.65	73.33	27.54	3.98	40.69	82.60	53.18	33.10
SP_{1_05}	**77.02**	**25.06**	**76.53**	**32.81**	**7.33**	**43.91**	84.63	**57.86**	**45.62**
Random SP_{2_1}	71.45	21.25	74.38	29.50	2.66	32.38	81.41	51.71	23.88
Random SP_{4_1}	69.32	20.25	67.38	13.98	6.53	30.26	**85.01**	49.23	27.68

Table 5.6: Test results on using larger pre-computed chunks for the random dynamic division method.

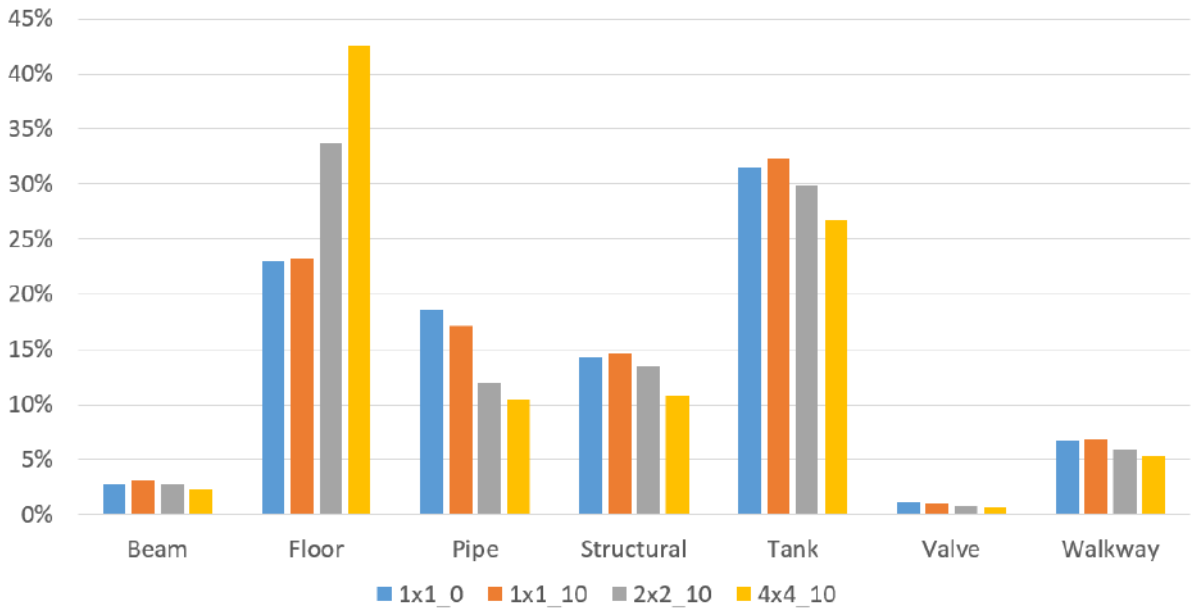


Figure 5.5: Semantic class distribution of the training dataset following different static division method.

this strategy with the larger pre-divided chunks (Tab. 5.6).

Both the overall performance (-1.0 mIoU) and by class performance decreases when increasing the pre-computed chunks size. The worst decrease concerns the Beam class (-15.52 IoU compared to SP_{2_1}) whereas the segmentation quality of the Tank and Walkway class is mostly conserved (-2.12 and -2.48 IoU respectively). Moreover, the performance on the valve, structural and floor increase, with Random SP_{4_1} becoming the best performer on the floor class. Those changes can alert the presence of a major shift in the effective balance between classes observed by the network during training.

The semantic class distribution of the training dataset, once divided in chunks by the static method, is presented in Figure 5.5. As each shape centre point during the division process is chosen randomly, bare the effect of variation in point density, every point should roughly have the same chance to be selected for training in the dataset. Thus, the significant increase in points representing the floor class is a major imbalance and is probably the cause behind the lack of performance of Random SP_{4_1}.

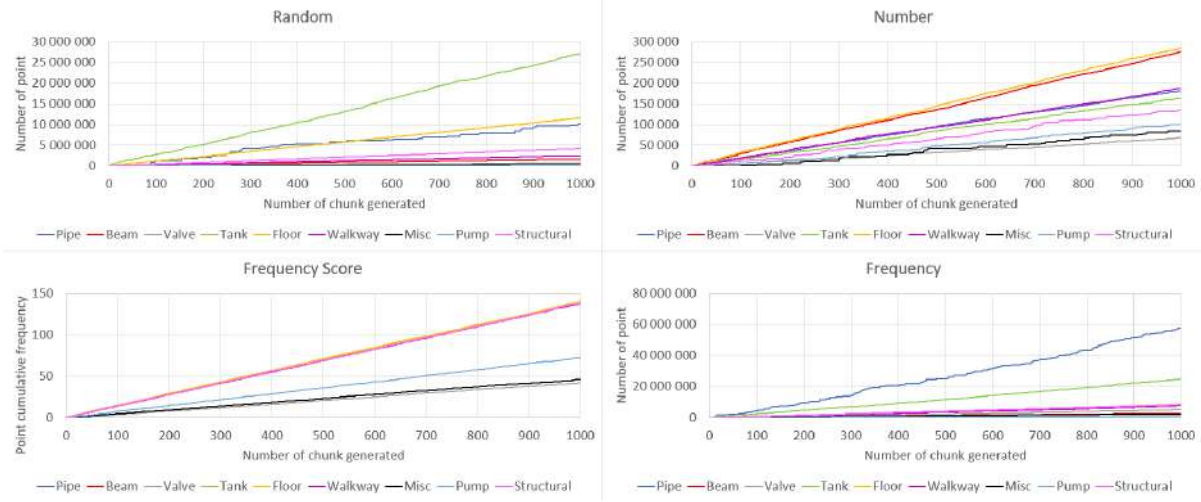


Figure 5.6: Number of point generated per class for the different strategy when SP_{4_1} is divided dynamically.

5.1.3.3 Reset

Working on improving the most promising drawing strategy, Frequency, seems to be necessary to enhance segmentation results. However, if the method is applied to SP_{4_1} directly, the predominance of point representing pipe worsen (Fig. 5.6). More points are sampled but the difference in number of points per class presented to the network increases. Thus, a first potential reason behind the lack of effectiveness of the method compared to a random division or SP_{1_05} is a gradual loss of meaning in using variance as a criterion to discriminate chunks. The data allocation is updated each time a chunk is drawn and never reset during training. As is visible in Figure 5.7, variance increases quadratically during training. It is thus possible that each new chunk has less and less influence on the method decision. A state of equilibrium is reached between the frequency of the major classes (Pipe, beam, tank, floor, walkway and structural) and disturbing this order does not immediately decrease variance. It is stuck in a local minimum. The method can be considered as a greedy algorithm and a common method used to alleviate this problem is to perturb the algorithm, such as with GRASP [35].

An idea to help the algorithm is to reset its saved data allocation regularly. In deep learning techniques, two events seem to be right for this reset: at the end of each batch and epoch. The two reset timing are considered and results are presented in Table 5.7. When a reset is done each epoch, its general performance is similar to the random strategy (+0.23 mIoU). The trade-off between the floor class and the valve class is still present compared to random (+4.71 IoU on valve, -4.61 on floor) and the segmentation of the Beam, Tank, Walkway and Structural classes improved. Resetting after each batch further improves the general performance of the method (+ 0.84 mIoU) and most classes but see a drop on the valve class (-6.17 IoU). Those results show an improvement but are far from sufficient compared to SP_{1_05} and SP_{1_0}.

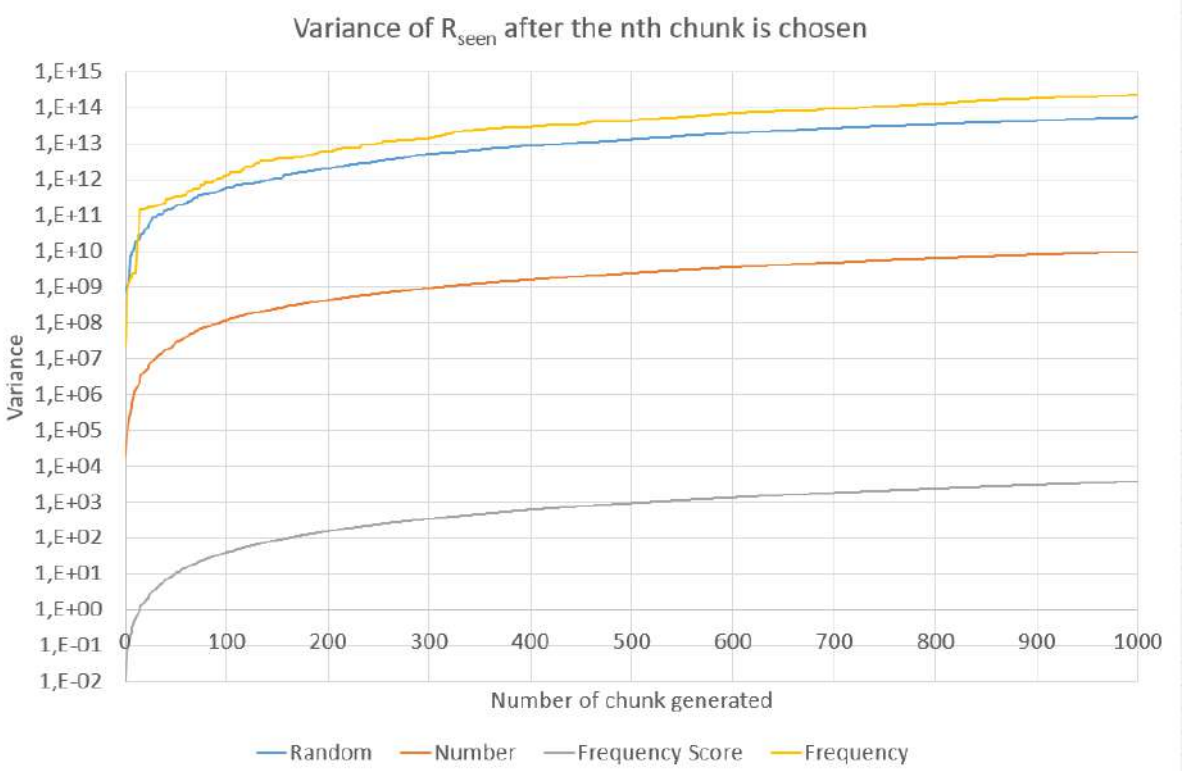


Figure 5.7: Variance of A_{seen} after the application of the division method.

Data shape	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
SP_{1_0}	73.17	22.65	73.33	27.54	3.98	40.69	82.60	53.18	33.10
SP_{1_05}	**77.02**	**25.06**	**76.53**	**32.81**	7.33	**43.91**	84.63	**57.86**	**45.62**
Random SP_{4_1}	69.32	20.25	67.38	13.98	6.53	30.26	**85.01**	49.23	27.68
Frequency Epoch SP_{4_1}	70.03	20.48	66.67	15.09	**11.24**	36.75	80.40	41.83	31.21
Frequency Batch SP_{4_1}	70.10	21.32	66.67	18.34	5.07	41.10	82.71	51.40	31.03

Table 5.7: Inference results obtained after modifying the Frequency division method.

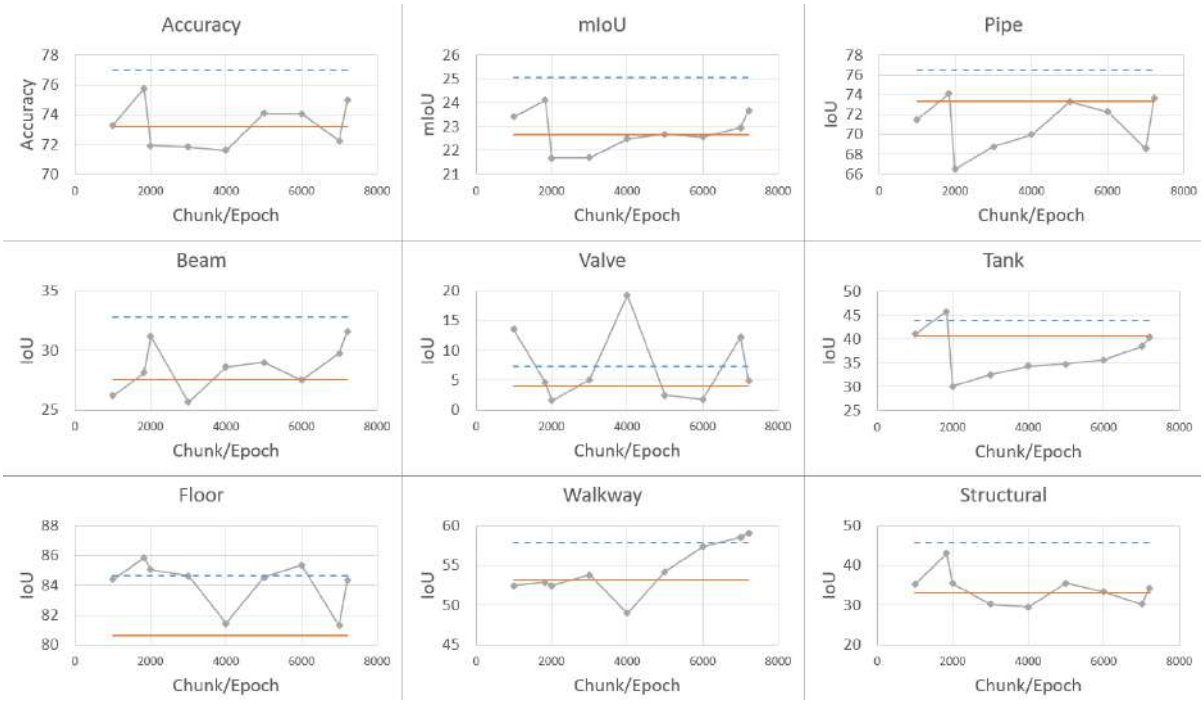


Figure 5.8: Segmentation results obtained after using different a different number of chunk per epoch during training. SP_{1_0} performance is represented by the solid line. SP_{1_05} performance is represented by the dotted line.

5.1.3.4 Random strategy on SP_{1_05}

From the previous experiments, it appears that one key problem to having the dynamic method working is the quality of the chunk computed. Those computed with the static method are more qualitative as they are extracted from whole scenes and then reduced. The risk of a chunk representing a smaller area than maximum chunk size is limited (SP_{1_0} mean surface is 0.947 m^2). A minimum balance and diversity guarantee is created as every chunk and data point has the same chance to be drawn with SP_{1_0}. Barring border points, the same is true for SP_{1_05}.

As seen previously, such properties are more complicated to guarantee when chunks are computed dynamically. The raw informativeness of the chunks computed with the static method (SP_{1_05}) is evaluated. The random method is used to draw a fixed number of chunk per epoch. Between 1000 and 7000 chunks are considered per epoch. Networks trained with the original size of SP_{1_0} (1829) and SP_{1_05} (7124) are also considered.

Looking are the results (Fig. 5.8), two cases can be considered: few chunks par epoch (<2000) or maximum chunks per epoch (>7000). Their relative success quite probably come from their relationship with the network learning-rate and its progressive reduction during training. Two regimes can be observed in Figure 5.9, the network trained with 7124 chunks per epoch progress towards a stable optimum after only 100 epochs whereas the network trained with 1000 chunks per epoch is less stable but quickly (in training time) obtains a good solution. One (1000) is forced to learn only the essentials as the learning-

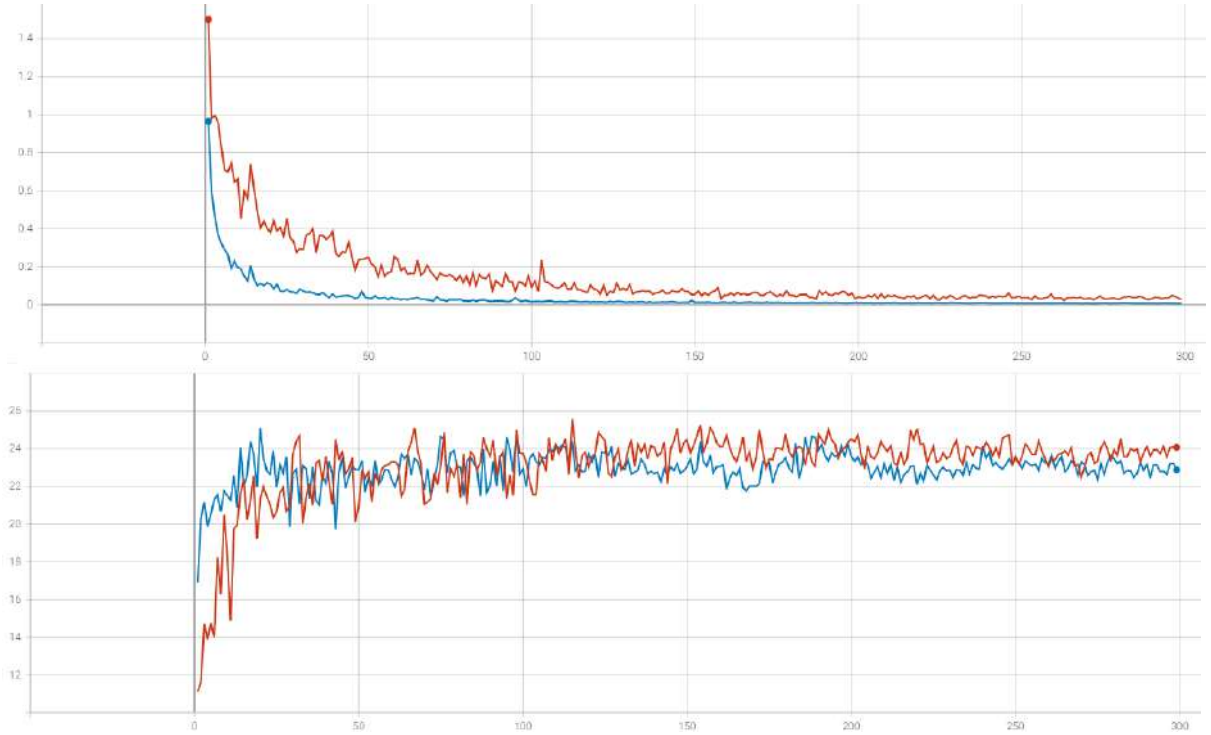


Figure 5.9: Evolution of the training loss (top) and mIoU on the testing dataset (bottom) during training, per epoch. The network trained with 1000 chunks par epoch is in red, the one with 7124 chunk per epoch is in blue.

rate-to-chunk-seen ratio diminish quickly, the other (7124) has more time to explore the solution space. As the general project linked to this thesis is still in its experimental phase, it seems more judicious to favour the quickest network at first (1000 chunks per epoch). In future works, once the PCSS solution will be more developed, a longer training time could then be considered.

5.1.4 Ratio

Following the previous experiments on dynamic scene division and the difficulties encountered when using the colouring method on industrial data in Section 4.3.2, a technique merging the two is created. The idea is that instead of using heterogeneous data to train a single colouring network, several colouring networks can be trained separately on homogeneous data. The process to determine those homogeneous subgroups of data can be done manually or automatically. For the SMARI and S3DIS [10], ways to quickly separate manually the data are available based on the content of those networks[2]. Envisaged, but not fully tested, automatic method includes the use of t-SNE to determine the grouping followed by a clustering method, such as K-NN. One reason behind the lack of experimentation on the automatic possibility is that this step is an opportunity to add informativeness to data at an extremely low manual cost: using the advice of an expert on

2. That is, acquisition location or room function, as will be seen later in the experimental part.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Colourless synthetic	73.17	22.65	73.33	27.54	3.98	40.69	82.60	**53.18**	33.10
Colour4	72.52	22.36	71.78	30.47	1.90	40.14	81.28	52.82	33.19
Colour3	73.64	22.97	72.61	29.02	**5.54**	41.51	82.94	50.27	38.07
Max_{C1}	72.82	22.41	73.50	31.16	3.77	43.90	**85.35**	50.19	35.93
$R = 0.25$	**74.44**	**23.50**	73.58	31.52	2.78	**45.50**	83.07	48.77	**42.02**
$R = 0.50$	74.26	23.48	73.69	31.88	4.15	43.84	84.83	50.22	38.06
$R = 0.577$	74.12	23.12	74.61	**32.82**	0.75	42.55	84.38	50.92	35.26
$R = 0.75$	72.82	22.60	**74.94**	29.04	4.23	43.43	84.15	48.75	30.71

Table 5.8: Inference results following the use of different ratio between real and synthetic data during training.

the dataset. Nonetheless, comparing the ability of an automatic method with the manual method could be done by repeating the experiments with t-SNE + K-NN generated subgroups. It is envisioned as a possible future work.

Once one colouring network is trained by subgroup, the synthetic data can be coloured by each method. This lead to a synthetic dataset several times larger and composed of several equivalent point clouds (Def. 3) with a different colour feature. By randomly drawing scenes, it is possible to keep a desired ratio of acquired to synthetic data seen by the network during training. In the following experiments, the ratio is the probability to draw a real data chunk instead of a synthetic one. This method is included in the paper accepted at the SPIE 2023 conference.

5.1.4.1 SMARI

This first experiment is carried on the SMARI dataset, as it was the one where the colouring failed to greatly improve segmentation quality (Sec. 4.3.2). To ease the comparison with experiments presented in Section 4.3, the data used are the same (the acquired SMARI dataset and Synth3). Chunk shape is also kept at SP_{1_0}. Four different colouring networks are trained on either ADP, DPO, KAM or YAR.

Different ratio are expressed during training. The results are reported Table 5.8. For comparison, results of Section 4.3.2 are also presented in the table. As the test focus on using data coloured with the network trained separately on ADP, DPO, KAM and YAR, the maximum results obtained with data colouring from those networks (Max_{c1}) are used as a basis for comparison. Those previous results are described in Table 4.15 and Table 4.16.

Mixing synthetic data coloured differently with the use of ratio improves the overall results: in all case, the mIoU is superior to Max_{c1} and, in three out of the four ratio studied, superior to Colour3. A ratio lower or equal than the original ratio of 0.577 appear to be the most beneficial to the network training. As the improvements are general and not focused on specific classes, it seems to confirm findings in Section 4.3: the colouring

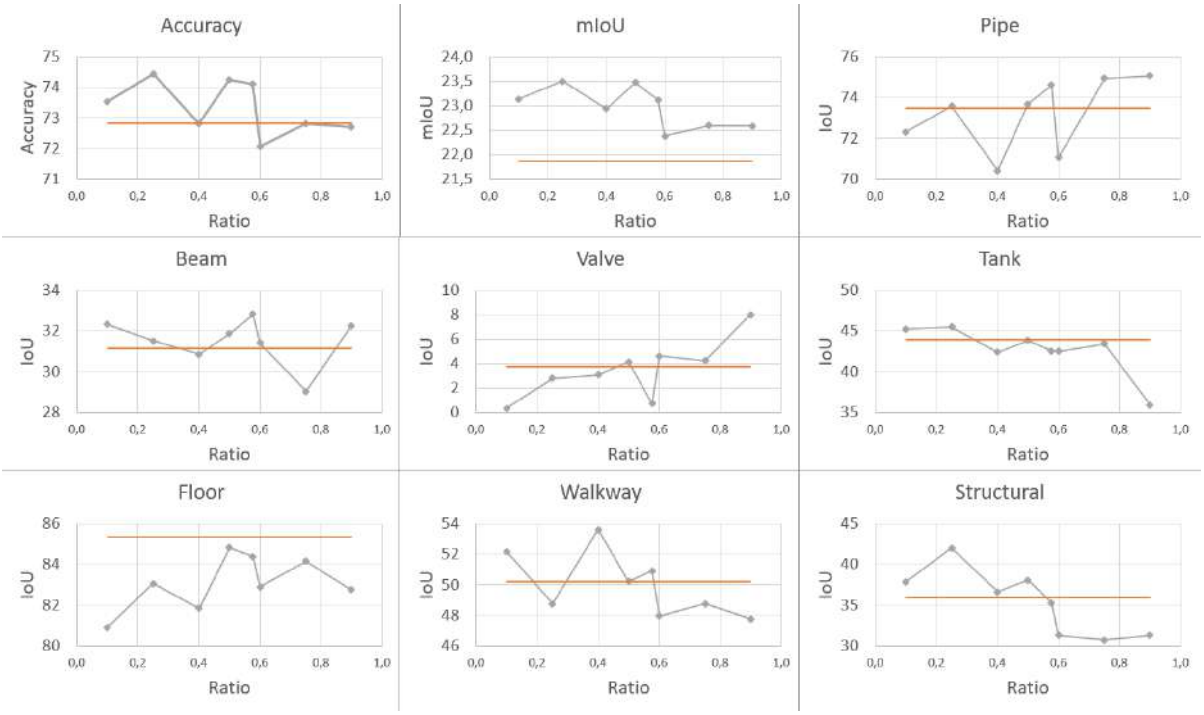


Figure 5.10: Differences in segmentation performance following the ratio of real to synthetic data used during training. Max_{c1} performance is represented by the orange line.

network transfers information to synthetic data. Needing to use a higher ratio of synthetic to acquired data could imply that knowledge present in synthetic data is more difficult to extract. Thus, when using the ratios techniques, it seems advantageous to use a ratio of real data between 0.25 and 0.5.

However, a direct link between ratio and class specific performance is more difficult to see. Searching for a trend, more networks are trained with different ratios. Performances by class are presented in Figure 5.10. In general, a peak in performance can be observed at 0.5. Moreover, if the mIoU is better in every case, there is always a trade-off occurring during learning as some classes tend to perform better in either the [0.1;0.4] range or the [0.6;0.9] range. Looking at the constant decrease in performance when segmenting the floor class, the ratio method may be useful to either counter class-imbalance or help a network which focuses too much on class represented by simple geometric shapes.

Following those results and the performance attained by SP_{1_05} in Section 5.1.2, one could think that the use of ratio is not necessary and that synthetic data could simply be added multiple times with different colouring. Such training is carried out and results are presented Table 5.9. From those, it appears that using multiple colouring network trained on different dataset is indeed more effective than mixing data during the training of the colouring network. The general performance of Colour4 is inferior to All 4 by 1.0 mIoU. Moreover, the performance of All 4 approaches the best cases when using ratio. Thus, this method can be seen as a useful baseline method when data available for colouring are highly heterogeneous. However, it should not be forgotten that All 4 used 4 times

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Colourless synthetic	73.17	22.65	73.33	27.54	3.98	40.69	82.60	53.18	33.10
Colour4	72.52	22.36	71.78	30.47	1.90	40.14	81.28	52.82	33.19
Colour3	73.64	22.97	72.61	29.02	5.54	41.51	82.94	50.27	38.07
Max_{C1}	72.82	22.41	73.50	31.16	3.77	43.90	**85.35**	50.19	35.93
$R = 0.25$	**74.44**	**23.50**	73.58	31.52	2.78	**45.50**	83.07	48.77	**42.02**
$R = 0.50$	74.26	23.48	73.69	31.88	4.15	43.84	84.83	50.22	38.06
$R = 0.577$	74.12	23.12	74.61	**32.82**	0.75	42.55	84.38	50.92	35.26
$R = 0.75$	72.82	22.60	**74.94**	29.04	4.23	43.43	84.15	48.75	30.71
All 4	73.93	23.36	71.12	30.17	**6.67**	41.66	82.15	**53.53**	39.88

Table 5.9: Inference results evaluating the necessity to use ratio between real and synthetic data during training.

Figure 5.11: T-SNE representation of the normalised colour histogram of each room contained in the S3DIS [10] dataset. Point colour represent area on the left image and room type on the right image.

more synthetic data than any ratios methods to bring similar results. As was seen in Section 5.1.2 and Section 5.1.3, the quantity of data and the number of batch per epoch used during training influence heavily on the network segmentation performance. Thus, ratio is a superior method even if it brings another meta-parameter to account for during training.

5.1.4.2 S3DIS

The conclusion on the superiority of the ratio method compared to using a single colouring network can be confirmed on the S3DIS dataset [10]. Contrary to our industrial case, the colours in S3DIS are more homogeneous from one zone to another (Sec. 4.3.2).

When looking at the S3DIS dataset, two possible divisions come to mind: dividing by

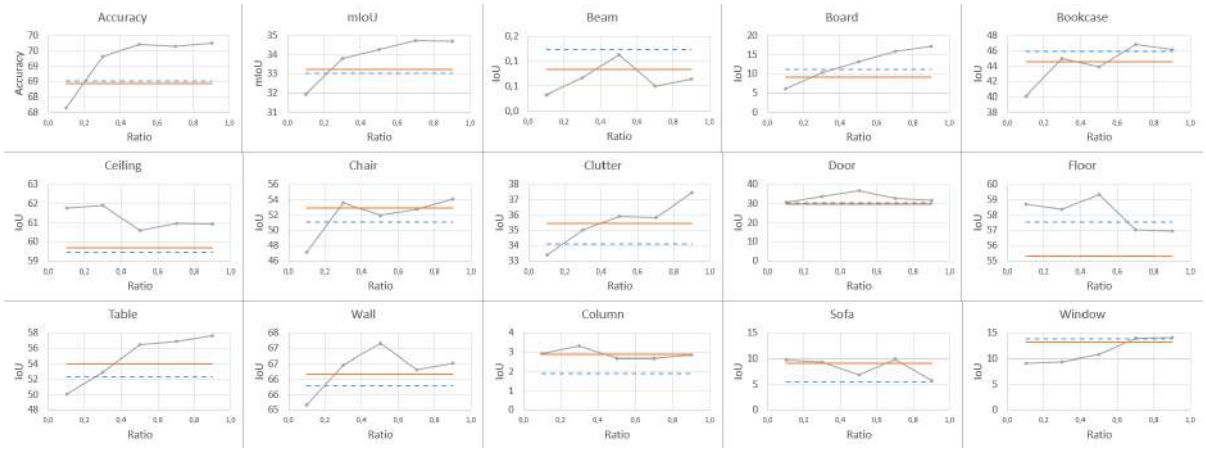


Figure 5.12: Differences in segmentation performance following the ratio of real to synthetic data used during training. *Off* performance is represented by the dotted line. *All* performance is represented by the solid line.

room type or area. If the colour histogram of each room is considered as a 768D point, it is possible to embed those points in a lower dimension with the t-SNE technique. When the histograms values are divided by the number of point they represent, the groupings presented in Figure 5.11 are observable. This corroborates with the observations made in Section 4.3.2: the colouration varies more by room function than by area.

Thus, instead of dividing the colouring dataset following the zone, it is divided following the room function. Nine types of room are considered. They are grouped in five categories to balance the quantity of data available to each colouring network during training:

1. Office
2. Hallway
3. Conference room
4. Auditorium, open space, lobby
5. Copy room, storage, pantry.

The same mix of acquired data and synthetic data as in Section 4.3 is used to train the semantic segmentation network. Two baselines segmentation networks are considered. The first follows the process used in Section 4.3.1. The synthetic data are coloured by a network trained on area 2, 3, 4 and 6 (*All*). A second semantic segmentation network is trained on data coloured by the network trained only on the office subset (*Off*). Finally, several networks are trained with synthetic data coloured by the 5 different colouring networks with different ratio. Figure 5.12 shows that this method also work on a more homogeneous dataset. The overall segmentation quality increases until a ratio of 0.5-0.75. This also hold for most classes, with only the floor and the ceiling seeing a decrease with a ratio of 0.75 and 0.9 (relative to results with ratio of 0.1 and 0.25).

Those results confirm the relevance of this colouring method, even when the data is

homogeneously coloured. The shift in optimal ratio (0.75 for S3DIS, 0.5 for SMARI) also hints at a difference in synthetic data quality between the two datasets. Manually crafted synthetic scene or CAD models are more informative than randomly generated scenes. This is coherent with the findings in Section 4.2.

5.1.5 Discussion

From the experiments carried out in this part, it seems that point density is much more important than data chunk shape, as long as the shape is pillar shaped. Removing the connection between the data chunk and the floor have an adverse effect in most cases. A simple shape, a square-based pillar, seems to be as efficient as others. Increasing the area covered by each chunk is a promising approach but should not be traded against a loss in point density or a lesser number of chunk par batch.

Trying to dynamically balance data by computing and selecting chunks each time new data is requested by the network during training is a complex matter. The current method does not succeed in besting the static method. However, its capabilities were increased following multiple experiments. Using ways to guarantee a minimum are per chunk could be key to make the dynamic method bests the static one.

Nonetheless, this study in dynamic data chunk computation allowed for the creation of an improved colouring method, which work both with our industrial data and the S3DIS [10] dataset. Moreover, how the way point clouds are divided before being fed to the segmentation network influences its capabilities is an understudied subject (Sec. 2.3.3.3). The conducted study creates a foundation onto which future knowledge can be built.

5.2 Data augmentation

The concept of data augmentation is not new in the context of deep learning, as was seen in Section 2.3.3.3. However, the efficacy of data augmentation methods is not always demonstrated in the literature. Even more so for PCSS for which no conclusive data could be found on their use in the specific context surrounding this thesis: semantic segmentation of point clouds representing an industrial scene. Two data augmentation techniques are studied in this part. The first one is a static approach where point cloud curvature is computed in an attempt to solve problems surrounding some classes of industrial objects (Sec. 5.2.1). The second approach is dynamic. It studies the effect of a rotation transformation on segmentation robustness towards variation on input data (Sec. 5.2.2). The successes and shortcomings of both methods are discussed at the end of this part in Section 5.2.3.

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Normal only	**72.94**	**22.50**	**72.21**	27.33	**4.11**	38.64	81.79	**51.38**	**36.83**
Curvature only	72.25	21.69	69.83	23.80	1.07	**42.65**	**82.14**	47.75	34.49
Normal + Curvature	70.47	21.29	67.35	**27.62**	3.71	38.83	80.48	49.40	28.36

Table 5.10: Results obtained by modifying orientation information given to the segmentation network input

5.2.1 Curvature

Testing the effect of precomputed data augmentation can be done by working on point cloud curvature. Curvature of a point cloud can be though as a measure of the roundness of the surface represented by the points. By default, our segmentation network uses as input points position, colour and normal. As an important number of industrial objects are cylindrically shaped, it can be interesting to add curvature information to the point clouds. Notably so when some recurrent problems with the segmentation results are considered, such as HVAC pipes which are not correctly recognised. The computation of point cloud curvature is resource intensive, roughly half a second per data chunk. As such, it is a good example of an interesting feature that can be pre-computed to augment data.

The computation of curvature is done with the cloudcompy python library using a Gaussian approach. For each point, its neighbours within a predetermined radius of 0.1 m are used to compute its local curvature. As a minimum of 6 neighbour points are required to compute curvature, this feature can be unavailable to some points. In this case, a default value of -1 is used. The same dataset as in Section 4.2.3.2 is used (SMARI acquired data and Synth3 synthetic data). For this experiment to focus mostly on the influence of the curvature information, data is made colorless. Training of the segmentation network is carried with both curvature and normal information but also with curvature only.

The results obtained after inference are presented in Table 5.10. Both the general segmentation performance and by-class segmentation are in favour of using normal as feature instead of curvature. The hypothesis that some pipes would be better understood by using curvature is not verified as the IoU on this class falls by 2.38 when switching normal for curvature. Nonetheless, a silver lining is the improvement on the tank class (+ 4.01 IoU). Those being large objects, having information on the neighbourhood comportment via curvature seems helpful for the network. Following those results, normal can thus be considered as a more informative feature than curvature for the network. This can be due to the network ability to extract more knowledge from this less processed information.

Adding both normals and curvature as features could seem to be a good idea in theory, as the two do not seem to influence the same classes. In practice however, the segmentation quality fall by 1.21 mIoU and the results are only marginally better for the

beam class. Those two conflict with each other by possibly giving too much information to the network.

Following these results and considering the time necessary to compute a point cloud curvature, the use of this curvature augmentation is not further researched.

5.2.2 Rotation

Industrial installations are man-made and follow the Manhattan hypothesis [29] in most cases. Most objects in a scene are aligned along two to three perpendicular axes. A constraint of this thesis is to create a method robust to rotation variation of the scene (Sec. 1.3). This is because, even if objects in the same scene are mostly aligned with each other, the coordinate bases of each scene are not aligned with each other. Even at the same general location, different parts of the industrial installation will be positioned differently. The *Piping* test scene represents the transition between two such locations, where the angle between some pipes is close to 140° (Fig. 4.6). In [61], Lin et al. showed that the semantic segmentation network currently used is not robust by itself towards rotation transformations. They propose quite successfully to use a new kernel which is more robust towards this kind of transformation. Here we pose the hypothesis that there is no need to alter the segmentation method and that robustness towards rotation can be learned by data augmentation.

This hypothesis is first verified before the possible implication of this augmentation towards data division are investigated.

5.2.2.1 Method validation

To achieve this a random division is applied to input data dynamically. Two versions of this augmentation are considered, one around the vertical axis and one around the three axes. In each case, one or more random angles are drawn uniformly in [0°;360°]. The synthetic data, Synth3, is added to the acquired data.

The network is tested on the usual dataset but also on rotated versions along the vertical axis. The range considered is [0°;345°] and a step of 15° is used between each version. The 0° version corresponds to the non-rotated testing dataset. Only the rotation around the vertical axis is considered as it is a constraint of the global project surrounding this thesis (Sec. 1.3). Training the segmentation network on data rotated around each axis is done to see if more information can be learned this way. If this proves true, this could be a sign that scene comprehension is orientation invariant with our chosen segmentation network.

The results obtained when testing on the non-rotated (Tab. 5.11) and rotated datasets (Fig. 5.13) agree on the superiority of using a rotation augmentation around the Z axis. Its overall performance is better than a network trained without augmentation (+ 1.09

Training	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
No rotation	74.39	23.19	75.22	**30.62**	**3.91**	38.82	**84.95**	53.94	35.34
Rotation Z	**76.31**	**24.28**	**77.02**	25.44	1.94	49.38	83.48	**54.86**	**44.45**
Rotation XYZ	62.64	18.01	53.30	16.97	0.07	**52.75**	66.62	48.21	13.51

Table 5.11: Results obtained by varying the rotation augmentation used.

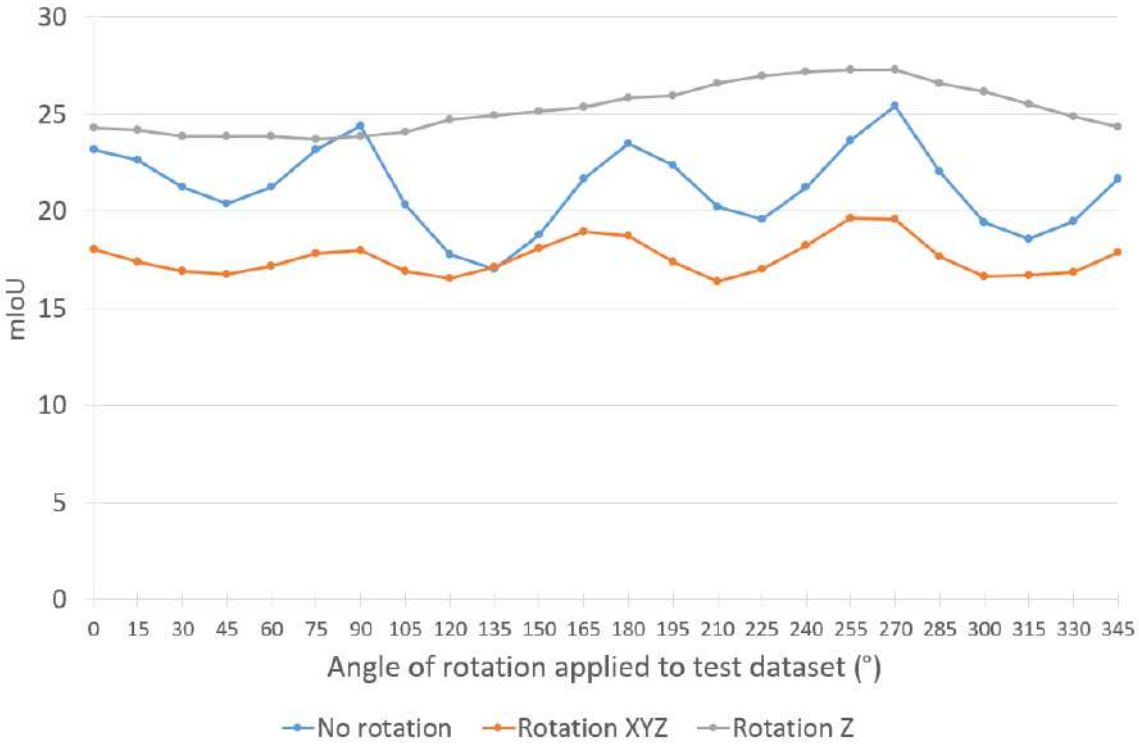


Figure 5.13: Inference results following the rotation augmentation used during training.

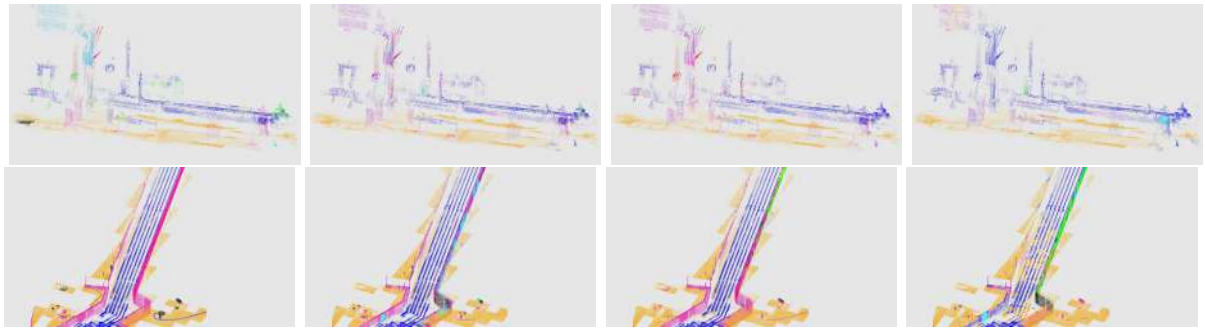


Figure 5.14: Inference results following the rotation augmentation used during training. From left to right: Ground truth, without rotation, rotation Z, rotation XYZ. *Unit2* scene on top row, Piping scene on bottom row.

mIoU) on the non-rotated dataset but also in almost every rotated case (except for 90°). The most improvements are observable on the structural (+9.11 IoU) and tank (+10.56 IoU) classes. These improvements come from two things: clearing up a misunderstanding between HVAC pipe and tank in the Plant scene and a better understanding of wall structures in the *Piping* scene (Fig. 5.14). The rotation augmentation allows the network to better understand highly spatially oriented elements.

This link between object orientation and semantic meaning can also be observed in the results on network trained with no augmentation. As can be seen in Figure 5.13, the segmentation network performs better on scene rotated by 90° and 270°. It seems that our test dataset main orientation is perpendicular to the training dataset main orientation[3]. The cyclical aspect of the result is also a strong clue on the important relation between object orientation and meaning. The cycle having a frequency of 90°, it contrasts with the results obtained by 3D-GCN [61] where increasing rotation only decreased performance and no cycle is visible in the [0°; 180°] range. This implies that the dataset they used to train the different models contained objects with standardised orientations[4].

The poor results obtained when using a three-axis rotation augmentation show that scene comprehension is not rotation invariant (+5.18 mIoU). However, it still has a regulatory effect when the test is carried out on rotated scenes: the maximum amplitude of the cycle is divided by 2.54. Its ability to comprehend pipe with very large bore (HVAC) is improved, which also increases its tank IoU by 13.93. However, as the relation to the floor is muddled by the augmentation, almost everything is understood as a pipe in the *Unit2* scene (-21.92 IoU, Figure 5.14) and the wall in the *Piping* scene is thought as floor (-21.83 IoU, Figure 5.14).

Following this experiment, the rotation augmentation is kept as part of the overall solution.

3. A visual inspection seems to corroborate partially with this observation; *Unit 1* is perpendicular to *Unit 2*, *Storage 1* and *Storage 2* are mostly perpendicular to *Piping*.
4. Experiences in working with the ModelNet [111] dataset confirms this.

Colouring	OA	mIoU	beam	board	book.	ceil.	chair	clut.	door	floor	table	wall	col.	sofa	wind.
SP_{05}	81.89	49.94	0.05	56.51	62.06	**83.47**	71.57	46.76	28.38	**90.75**	65.18	74.08	**10.30**	26.51	33.57
SP_{10}	**83.03**	**53.17**	0.05	**60.15**	**63.72**	82.65	77.33	**49.68**	**39.34**	87.27	**72.71**	**75.44**	9.77	31.84	**41.21**
SP_{15}	79.66	50.48	0.03	57.75	62.89	70.84	77.67	48.42	32.44	77.04	72.58	74.29	9.04	33.28	39.93
SP_{20}	80.46	50.60	0.00	53.72	61.89	73.60	**77.96**	48.40	33.55	78.93	71.34	74.65	6.70	**37.60**	39.46
SB_{10}	77.57	43.82	0.04	28.69	57.28	**80.94**	71.06	43.14	19.55	**84.64**	60.59	67.97	6.72	16.58	32.40
SB_{15}	77.88	48.15	0.03	54.41	59.88	67.76	76.71	46.52	30.19	75.58	63.87	73.35	7.05	31.06	**39.51**
SB_{20}	**78.08**	**48.86**	**0.09**	**56.69**	**60.58**	67.24	**76.53**	**46.66**	**32.38**	75.11	**65.91**	**74.10**	**8.86**	**31.73**	39.23

Table 5.12: Comparison of the effect of chunk size on a network trained on the S3DIS dataset.

5.2.2.2 Implications towards data division

To verify if the rotation augmentation has a positive or negative influence on chunk shape, an experiment is conducted on the S3DIS dataset [10]. For this experiment, only acquired data are used: areas 1, 2, 3, 4 and 6 for training the segmentation network and area 5 to test it. Two shapes are considered: a ball-sphere as in the original article [92] and square-based pillars. The room division proposed for the S3DIS dataset is used in place of the static division algorithm and the training last for 300 epochs. In all cases, only the rotation around the vertical axis is applied.

First, the chunk size parameter is evaluated; 3000 and 2000 chunks per epochs are used in the pillar and sphere cases respectively. The size used are chosen following observations made in Section 5.1. For pillar, sides of 0.5, 1.0, 1.5 and 2.0 meters are used. Radius of 1.0, 1.5 and 2.0 meters are considered for the sphere. The results are presented in Table 5.12 and strongly favour the idea of optimally sized chunks. In both the pillar and sphere cases, the smallest size focuses more on the ceiling and floor than the others do. The network trains with denser but less diverse information and focuses on classes whose objects are easy to discern by shape even with a lack of context. The optimally sized chunks (SP_{10} and SB_{20}) seem to possess the best ratio of context to point density for the network during training. Most classes are well segmented. The only exception is the sofa class with SP_{10} for which a larger chunk size definitely seems necessary (+5.76 IoU on sofa for SP_{20}). The optimal radius of the sphere chunk being 2 m, it is consistent with the findings of the original article [92].

Now that the optimal size of chunks is verified for the S3DIS dataset, the two shapes can be compared with and without the augmentation. In order to do so, we follow the original article and extract 5000 chunks per epoch. From the results presented in Table 5.13, it appears that a pillar shaped chunk is also superior to a sphere shaped chunk, even when changing the dataset and adding a rotation augmentation.

Be it with or without the augmentation, the differences of results between SP_{10} and SB_{20} are extremely similar. In most cases, SP_{10} is superior to SB_{20} and the classes where the method is superior are the same in both cases. The only exception is the sofa class where SB_{20} gains a small advantage (+ 0.95 IoU) compared to SP_{10}.

From these results, it appears that the rotation augmentation does not have a large

Colouring	OA	mIoU	beam	board	book.	ceil.	chair	clut.	door	floor	table	wall	col.	sofa	wind.
Without the rotation augmentation															
SP_{10}	**82.24**	**51.06**	**0.04**	**55.78**	**61.05**	**83.34**	**78.29**	**47.85**	**30.93**	**89.21**	**71.75**	73.94	9.82	**28.32**	33.44
SB_{20}	77.58	47.01	0.02	52.58	57.49	66.93	76.42	45.06	28.44	75.77	63.25	**74.10**	**11.28**	24.04	**35.78**
With the rotation augmentation															
SP_{10}	**83.71**	**53.55**	**0.48**	**63.00**	**63.42**	**85.01**	**77.81**	**51.21**	**36.98**	**89.87**	**72.47**	75.36	8.38	34.41	37.82
SB_{20}	80.08	50.84	0.00	61.30	61.48	72.47	76.91	49.55	29.10	78.14	70.34	**75.38**	**9.33**	**35.50**	**41.39**

Table 5.13: Comparison of the effect of chunk shape on a network trained on the S3DIS dataset.

impact on the influence over segmentation quality that hold chunk shape. A secondary, but surprising, result is the advantage of SP_{10} over SB_{20}. The article presenting the use segmentation method used SB_{20} has a chunk shape [92]. That using SP_{10} offers better segmentation results is unexpected. A possible link between this result and the absence of use of the inference strategy proposed by Thomas [92] could be a possible reason.

5.2.3 Discussion

In this section, two augmentation techniques were tested, one static and one dynamic. The goal of each augmentation was to solve a problem related to our application of PCSS to the industrial domain.

The static augmentation, adding curvature as an input feature, failed to improve the network segmentation abilities. It nonetheless showed the advantages in using a less processed feature that can also be precomputed: point normal vector coordinates.

The dynamic augmentation, computing a random rotation around the vertical axes for each chunk, proved useful by giving the segmentation network robustness towards rotation. It also improves the network segmentation ability as a whole. As such, the augmentation technique is kept. Considering this augmentation ability to improve segmentation quality, investigating the influence of other isometric transformation (translation and symmetry) using a similar methodology of experimentation could prove valuable. If, following the results presented by Lin et al. [61], translation would prove to be less valuable for the segmentation network used in this work, it should work for other methods, such as PointNet [75]. Symmetry is an augmentation technique often used in 2D images but whose use is less documented for PCSS (Sec. 2.3.3.3).

5.3 Data classification

In this section, the classification used for PCSS of industrial data, defined in Section 3.3.1, is questioned. In order to find a more consistent classification, an approach is first proposed in Section 5.3.1 before experiments are used to select one of the classification proposed (Sec. 5.3.2).

5.3.1 Methodology

A new classification of data is first proposed. It is defined by several subclasses that can be moved between the proposed classifications. As analysing results obtained on the same but differently classified data can be difficult, methods to compare the different results are then described.

5.3.1.1 Considered classifications

To construct a new classification, flaws in the original one must be examined to ascertain how the classes must be reorganised.

Piping In most experiments, the segmentation of pipes is quite good but not excellent (around 70 IoU in most cases). This class is also considered important (Sec. 3.1). As such, the previous pipe class must be kept. Another important component of industrial settings are valves, for which segmentation is more difficult (IoU is less than 10 in most experiments). As such, the two classes can be combined in a unique Piping class. Also included in this piping class is PipingRack, for which segmentation failed completely.

Metallic The segmentation of the beam class depends entirely on synthetic data. When enough is added, significant progresses are observed. This class is often mistaken with the structural class. As the difference between the two is blurry, it makes sense to group the two of them together. After all, structural is defined as "Structural element excluding beam. Architectural element such as wall, foundation and ceiling." (Sec. 3.1). Even if this definition reflected the focus of industrial draughtsmen, it is insufficient to train a neural network. The metallic elements part of this class are put together with the beam class to create a Metallic class. Those elements are small beams and miscellaneous metallic supports for pipe and equipment. Finally, even with its high shape complexity, the walkway class is often well segmented (around 50 IoU in most cases). As it is made of structural elements, even if they are assembled in a particular way, this class is also fused with the new Metallic class.

Equipment When looking at the definition of equipment by Agapaki (Tab. 3.1), objects of this category are not treated the same by the network. There is a gap between the tanks, which the network segments relatively well, and pumps that are not recognised. To simplify the definition of equipment for the network, it will be defined as everything connected to the piping system. Thus, this comprises the tanks and the pumps but also instrumentation.

Architecture The decision to extract the metallic structural elements from the original structural class left a few kinds of object behind, notably concrete blocks used as support

for piping and metallic elements. Those elements are put in an architecture class with the other elements defining a building architecture: wall, ceiling and floor. Thus, this class contrasts with the metallic class. Architecture is linked to the general scene organisation and represents elements of general context, whereas the metallic class is more linked to the elements of the scene and provides context on a smaller scale.

Electricity The electric system is not a priority target for the segmentation network (Sec. 3.1). Nonetheless, electric cable and electrical panel instances occur in the dataset which gave them a place in the original classification. As both the piping system and the structural system were defined as a single class, the electrical classes are also grouped in a single electrical class. In the original classification, each of the electrical objects were not recognised by the network, thus even a minimal segmentation of this class would be a significant improvement.

Miscellaneous With the same objective to simplify class definition for the network, the miscellaneous pieces of equipment which are too specific to be identified or do not represent an interest to the general project are put together with artefact in a single class.

This new classification, designed as S1, is illustrated in Figure 5.15. It includes subclasses that can be used to modify this new classification easily. Four other classifications are created from S1. The first isolates the floor from the architectural class to balance the number of points in the classes. It is named as S2. The second use S2 as a basis but also isolate tank from the other kind of equipment as it represents most of the points of this class but also represents objects extremely different from the others in terms of size. This is S3. Finally, S4 and S5 are based on S3 and tries to balance the equipment class. S4 do so by transferring the valve from the pipe class to the equipment class whereas S5 fuse the equipment class with the piping class.

Those new classifications create datasets that are slightly more balanced (Fig. 5.16), even if some classes are naturally more uncommon than others, such as electrical objects.

5.3.1.2 Comparison method

Due to the major difference in data distribution between those classifications, a methodology to compare each of their relative segmentation quality must be devised.

Three comparison strategies are possible: using overall accuracy, adjusting IoU per class and looking at the change of prediction veracity per point.

Using accuracy is the simplest method that provides a global view of the general segmentation quality. As it looks at the total of correctly classified points over the total

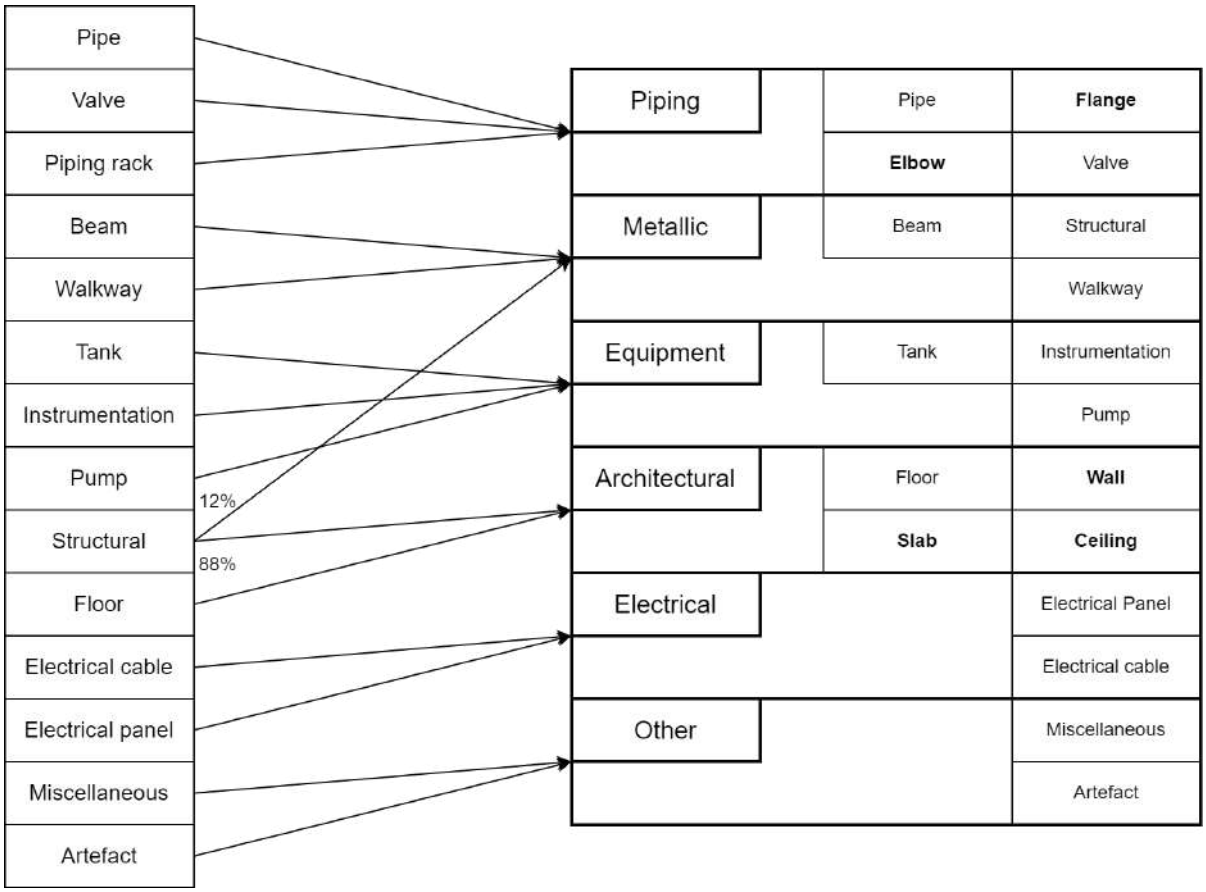


Figure 5.15: Illustration of how the new classes are constructed for S1. The new subgroups are indicated using a heavier font. The percentages of training points of the original structural class going to the metallic or architectural groups are included.

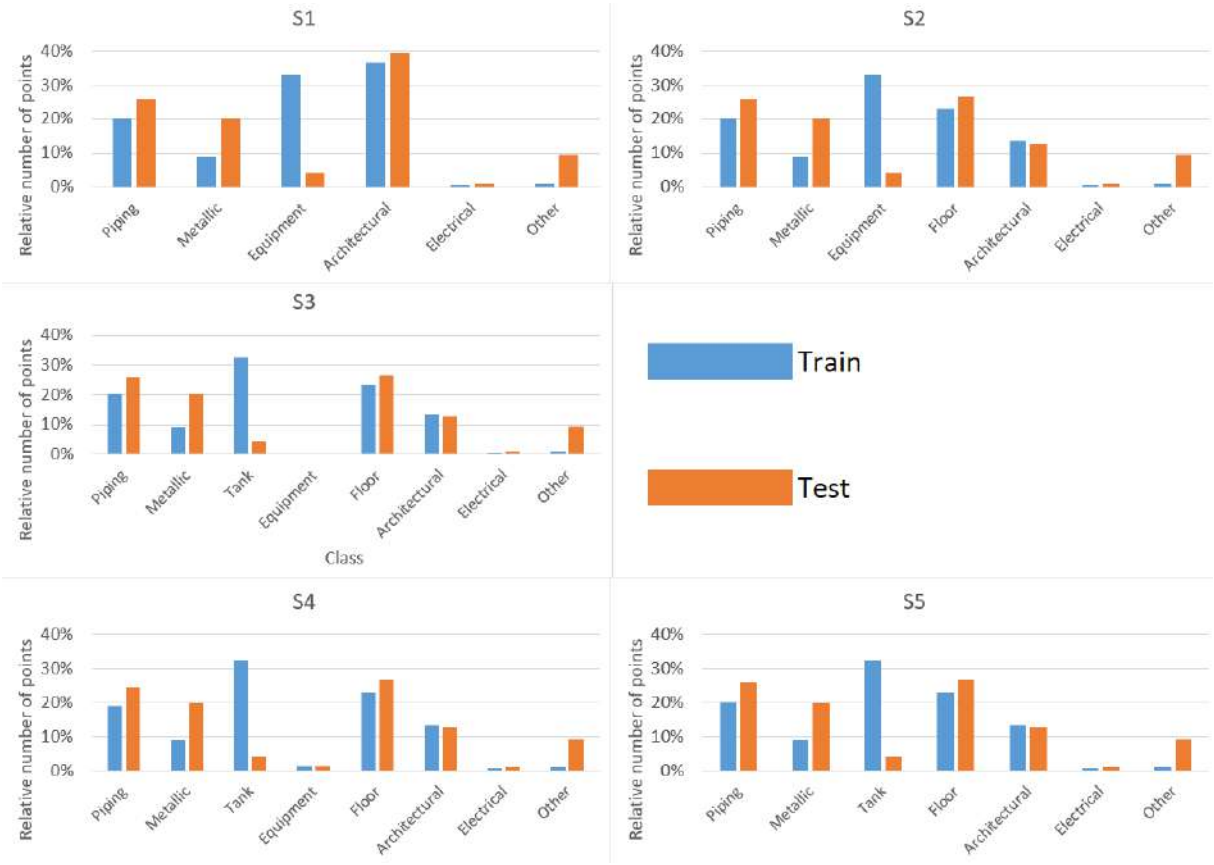


Figure 5.16: Data distribution between the different semantic labels of the new classifications. Distribution presented after reduction on SP_{1_0}.

number of points (as defined in Definition 5), it only offers an imperfect view of the results which is particularly influenced by class imbalance [71].

Adjusting the IoU per class can be made by computing the contribution of each class relative to one another. For example, when considering the test dataset, the S1 architectural class is composed of 67.5% floor and 32.6% structural classes from the original dataset. Thus, the adjusted IoU for the architectural class of a result obtained on the original dataset could be:

$$IoU_{architectural} = 0.675 * IoU_{floor} + 0.326 * IoU_{structural}$$

A problem with this measure is its interpretation. Should the ratio be computed using the test dataset or the training dataset? Does it create a reasonable approximation of the network performance? No equivalent seems to exist in the literature. As such, the effectiveness of this measure will be evaluated with the help of the other two methods. An apparent shortcoming of this technique is when a new class is only a subset of the original class, such as Architectural in S2. In this case, the comparison does not feel fair as it only partially represents the truth.

Looking at the change in prediction per point can be done on the point cloud saved after each inference. For the same point, the change in prediction can be evaluated, with only four cases possible: staying true (ST), staying false (SF), becoming true (BT) or becoming false (BF). This change in prediction can be evaluated by class to see if the change in classification has a different impact depending on the class considered. A rate of change can also be computed to evaluate the improvement on the total testing dataset or per class. This rate of change is computed as:

$$R = \frac{BT - BF}{ST + SF + BT + BF} \tag{5.4}$$

However, the computation of this metric is imperfect as only 4096 randomly drawn points are used in each data chunk (Sec. 3.4.2). Thus, only general trends can be determined when looking at this metric. As three networks are trained for each result presented (Sec. 3.4.3), the rate of change computed is obtained after cumulating the change of prediction per point on each combination of original and new classification.

5.3.2 Experiments

A first experiment is made to determine the value of new classifications (S1, S2 and S3) relative to the original one. It uses the acquired SMARI dataset and Synth3. The case with and without synthetic data augmentation (adding Synth3) are considered. When Synth3 is used, the Colour3 colouration network is used to apply colours, as in Section 4.3.2. In all cases, the coefficients used to adjust IoU are computed on the testing dataset.

Training	Acc	mIoU	Piping	Metallic	Equip.	Arch.	Elect.	Other
Original classification								
Acquired only	69.30	30.54	60.64	21.31	31.42	64.80	0.00	**5.08**
With synth. data	73.64	35.83	**66.01**	38.94	**40.39**	68.32	0.04	1.30
S1 classification								
Acquired only	70.96	34.51	57.76	49.37	27.51	68.89	**1.61**	1.92
With synth. data	**74.23**	**36.50**	63.45	**50.81**	29.28	**73.52**	0.89	1.01

Table 5.14: Comparison between networks trained on the original dataset and the S1 classifications.

Training	Acc	mIoU	Piping	Metallic	Equip.	Floor	Arch.	Elect.	Other
Original classification									
Acquired only	69.30	32.73	60.64	21.31	31.42	82.55	28.11	0.00	5.08
With synth. data	**73.64**	**38.24**	66.01	38.94	**40.39**	**82.94**	**38.07**	0.04	1.30
S2 classification									
Acquired only	70.15	36.79	60.98	45.71	33.96	81.03	27.85	**2.80**	**5.20**
With synth. data	71.86	36.34	**67.53**	**47.14**	28.39	82.77	27.43	0.24	0.89

Table 5.15: Comparison between networks trained on the original dataset and the S2 classifications.

Training	Acc	mIoU	Piping	Metallic	Tank	Equip.	Floor	Arch.	Elect.	Other
Original classification										
Acquired only	69.30	28.79	60.64	21.31	32.29	**0.31**	82.55	28.11	0.00	**5.08**
With synth. data	73.64	33.62	66.01	38.94	**41.51**	0.10	82.94	**38.07**	0.04	1.30
S3 classification										
Acquired only	71.45	32.41	66.89	43.34	33.62	0.00	**83.41**	26.09	1.48	4.44
With synth. data	**74.23**	**36.16**	**70.36**	**50.01**	41.45	0.02	81.59	37.25	**7.57**	1.07

Table 5.16: Comparison between networks trained on the original dataset and the S3 classifications.

	Acquired Only					With synthetic data				
Class	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
Pipe	-14.67	-10.29	2.47	-8.01	-3.24	-15.84	-4.99	-4.19	-1.04	4.87
Beam	31.30	18.00	15.95	20.98	16.60	23.27	15.22	18.01	22.12	20.72
Valve	75.95	72.14	64.10	6.39	57.62	59.13	56.07	59.38	18.57	54.64
Tank	-9.28	-2.36	-0.27	-2.16	-1.76	-1.43	0.58	-5.08	-2.97	-3.49
Elect. Cable	1.11	8.54	10.59	0.25	0.32	0.21	0.35	3.26	0.11	0.03
Floor	-1.13	0.01	2.88	2.52	0.41	3.23	-0.94	-3.55	-0.76	-1.03
Walkway	35.82	24.91	21.30	25.75	31.20	-2.82	-4.59	10.55	13.80	8.76
Misc.	-6.39	-1.55	-3.24	-3.09	-2.38	-1.31	-1.20	-0.63	-1.25	-0.30
Artefact	6.16	8.00	8.99	11.52	4.08	1.00	0.55	1.85	2.26	0.30
Piping Rack	0.82	1.27	0.97	0.83	0.30	0.35	0.13	2.33	0.12	0.18
Structural	6.58	-5.01	-8.35	-14.81	-9.20	1.95	-15.35	-4.95	2.80	4.47

Table 5.17: Rate of change (Eq. (5.4)), in percent, compared to the original classification.

The obtained accuracies seem to point in favour of the new classifications (Tab. 5.14, Tab. 5.15 and Tab. 5.16). The gains of segmentation quality per class observed with the adjusted IoU are explainable by the rates of change displayed in Table 5.17. Except for S1, the new classifications improve the network ability to detect piping. This can be explained by a small loss on pipe segmentation ability in exchange for a large gain in understanding valves ($R > 55\%$ on S1, S2 and S3). The same observation can be done on the metallic class. Grouping similar classes improves the network ability to understand said classes as a whole as it does not need to distinguish the details necessary to separate similar classes. However, this is also followed by a loss of precision in the class which is the most represented in the data. The rate of change is negative for pipe and floor on S1. When the floor is separated for the architecture class (S2, S3), it is the structural class which suffers.

One success of this method are the gains obtained on the electrical objects. If an IoU of 7.57 (S3) is still insufficient, it is a consequent improvement compared to the original classification.

This experiment shows that grouping classes for which data quantity is lacking improves the network ability to segment them, even when they are represented by largely different geometrical shapes (such as electrical panels and cables). Their semantic meaning should be key in explaining this phenomenon, even if it is currently difficult to test. Those results are in favour of S3, which in turn justify the creation of S4 and S5, two possible ways to further the grouping of similar classes. Looking at the results (Tab. 5.18 and Tab. 5.19), they are in favour of S5 for which accuracy is superior in both cases (with and without synthetic data). Their segmentation ability of most classes is also similar, with an exception for the architectural class, where S5 is ahead by 5.72 IoU when synthetic data are added. This lead more certainly come from the grouping of piping and piping-related equipment in a single class.

Training	Acc	mIoU	Piping	Metallic	Tank	Equip.	Floor	Arch.	Elect.	Other
Original classification										
Acquired only	69.30	29.83	63.80	21.31	32.29	**5.53**	82.55	28.11	0.00	**5.08**
With synth. data	73.64	34.68	69.49	38.94	**41.51**	5.17	82.94	38.07	0.04	1.30
S4 classification										
Acquired only	69.80	32.21	62.87	46.22	32.86	4.51	82.16	22.87	**1.83**	4.36
With synth. data	**75.11**	**36.24**	**70.91**	**50.65**	37.38	4.89	**83.38**	**40.94**	0.28	1.48

Table 5.18: Comparison between networks trained on the original dataset and the S4 classifications.

Training	Acc	mIoU	Piping	Metallic	Tank	Floor	Arch.	Elect.	Other
Original classification									
Acquired only	69.30	32.82	60.38	21.31	32.29	82.55	28.11	0.00	**5.08**
With synth. data	73.64	38.36	65.72	38.94	**41.51**	82.94	38.07	0.04	1.30
S5 classification									
Acquired only	70.72	36.95	64.21	47.04	36.34	81.23	25.66	**0.49**	3.68
With synth. data	**76.84**	**41.84**	**72.02**	**48.68**	39.98	**84.44**	**46.66**	0.14	0.96

Table 5.19: Comparison between networks trained on the original dataset and the S5 classifications.

Thus, grouping seems to be a good solution in order to help the network in understanding the general composition of an industrial installation. If it can be seen as an act of reducing segmentation difficulty, it is also a necessary step to move forward a hierarchical classification of industrial data. This will be an important step in the global project (along with going from a semantic segmentation to a panoptic segmentation). As such, the S5 classification is the one kept for the final experimentation carried out in Chapter 6.

5.4 Conclusion

In this chapter, three works surrounding the informativeness of data in general were presented. By working on the division process applied to point cloud data before it is fed to the network, a simple static division methods was selected. This work also showed that following the shape and size of data chunk, segmentation results were modified. However, it seems that it is more the density of point per chunk and the chunk size that are at play that its shape. The only influence of the shape is linked to data division along the vertical axis. When the dataset layout is vertical, sphere or cube chunks are less informative than pillar. This effect is more pronounced with the industrial data than with S3DIS [10] for which point clouds are rarely higher than 2 meters and the ceiling class can play the same role as the floor. Understanding the link between the floor and other semantic class seems to be necessary knowledge for the segmentation network. This idea could be further tested, either by conducting similar experiments

where the floor is removed from the dataset or creating a dataset where whole scenes are rotated randomly. With the first type of experiment, the difference between pillar and sphere should lessen for outdoor data and may increase for indoor data. With the second, the relative order between points is lost. There is no more a guaranteed layer of point representing objects above a layer of points representing the floor. In this case, a similarity in efficiency between pillar and sphere could be expected. This work could also be extended to other segmentation method and dataset in order to better identify the link between data division and segmentation performance. Datasets in other domains presents different constraints, such as SemanticKITTI [14] which is less dense, in points and objects, than our case studies. This could possibly lead to the necessity to work on larger data chunks. Segmentation networks based on MLP looses some geometric information when extracting features, which could lessen the differences between pillar-shaped and ball-shaped data chunks.

The work on division also allowed improving the colouring method. By using different colouring networks, trained on data following their acquisition location, multiple set of differently coloured synthetic data can be generated. Mixing those synthetic data randomly during training time is more advantageous than using a single colouring network as in Chapter 4.

The efficiency of a simple rotation augmentation was also verified. This augmentation, applied dynamically on each data chunk before the network input, makes the segmentation network robust towards rotation along the vertical axis but also improves its general capabilities.

Finally, the data classification relevance was explored. A new classification was proposed. This one simplifies the labelling of data for the network by considering more general classes and reflect the different system used in industrial installation (piping system, structural elements, electrical system...). Its underlying use of subclasses is also a preparatory work before exploring the possibility to use a hierarchy of classification when training the segmentation network.

Chapter 6

Validation

This chapter concludes the experimental work made in this thesis by describing the retained solution following the tests made in previous chapters. The overall solution is tested by using a larger dataset than in the other parts. The performance attained, relative to the same semantic segmentation network using acquired data only, shows the effectiveness of the method developed in this thesis (+13.28 mIoU).

Before being able to conclude on the work presented in this thesis, the data processing framework proposed in Chapter 3 must be revised by drawing on the conclusions reached from the experiments made in Chapter 4 and in Chapter 5.

6.1 Method

The method retained, presented in Figure 6.1, contains almost the same synthetic data processing method as described in Section 4.1. The Virtual Laser Sampling technique developed produces synthetic data with geometric characteristics similar enough to real data. VLS is used with the devised virtual laser positioning method, visibility reduction (Def. 8) included. However, the simple colouring method as presented in Section 4.1.3 is not used as it is. Instead, the colouring based on homogeneous subgroups is used (Sec. 5.1.4) with a ratio of 0.5. The data is divided into chunks (SP_{1_05}) statically but not dynamically. However, to reduce the training time, only a thousand chunks are drawn per epoch (Sec. 5.1.3.4). Those chunks contain colour and the point normals as features in addition to point coordinates. The rotation around the vertical axis is dynamically applied as an augmentation, as in Section 5.2.2.

Concerning data, the acquired SMARI dataset is used in combination with more synthetic data than usual. A total of 19 860 synthetic data chunks are used, including Synth3 and the other scenes designed for the experiments in Section 4.2 (KAL excluded). Other synthetic data, based on CAD models, are also used. The classification used is S5

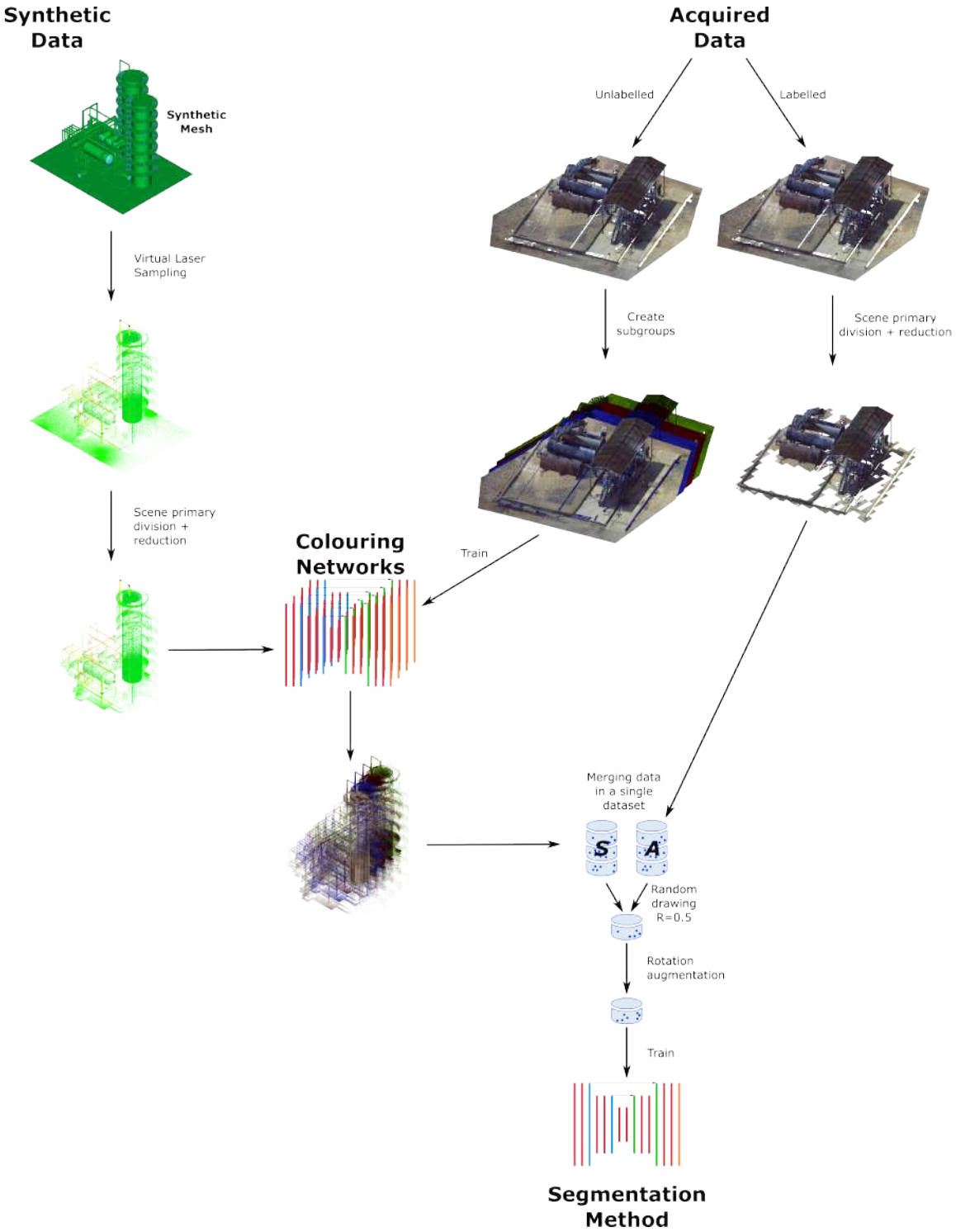


Figure 6.1: Finalised data processing workflow proposed in this thesis.

Method	Acc	mIoU	Piping	Metallic	Tank	Floor	Arch.	Elect.	Other
Acquired only	70.72	36.95	64.21	47.04	36.34	81.23	25.66	0.49	3.68
Simplified	76.84	41.84	72.02	48.68	39.98	84.44	46.66	0.14	0.96
SP_{1_0}	74.65	41.88	64.63	42.03	46.54	83.14	51.78	1.10	3.91
Not reduced	75.63	42.57	61.64	43.22	46.95	82.66	58.34	0.00	**5.21**
Complete	**83.14**	**50.23**	**79.73**	**58.13**	**57.06**	**87.52**	**65.52**	**1.80**	1.83

Table 6.1: Results obtained by the method devised during this thesis, using classification S5.

(Sec. 5.3).

6.2 Experiments

The parameters used are those originally described in Section 3.4. The learning rate is 0.005 initially and the first downsampling radius is 0.03 m. There are 16 chunks per batch and the network is trained for 300 epochs. The three networks took 12 hours to train, in total, on Computation machine 2. Having such a short training time can be an advantage when considering the current phase of the project containing this thesis. It allows a greater flexibility when experimenting on the subsequent parts of the SMARI solution.

Four methods are used as points of comparison, including the two used in Section 5.3.2. The first one, acquired only, is a straightforward application of the segmentation network to only the acquired data from the SMARI dataset. It represents the results obtained at the beginning of the thesis. The second method, simplified, corresponds to the results obtained with synthetic data augmentation only, as used in Chapter 4. The colour3 colouring network is used to process the synthetic data. In both of those cases, the data chunks are pre-computed and of shape SP_{1_0}. The SP_{1_0} method is an enhancement of acquired only using the results from Chapter 5. It uses the static data division without overlap as well as the random drawing. Like the overall solution, it is only trained with 1000 chunks per epoch. The last method is a downgraded version of the overall solution. Is it identical in all aspect, except that its data was not reduced (pre-computed chunks containing only the floor were not removed).

6.3 Results

The quantitative results are exposed in Table 6.1. Results obtained with the overall solution but using the initial classification are presented in Appendix B, Table B.3.

The large improvement in mIoU compared to either a network trained on acquired data only (+ 13.28) or with a simple synthetic data augmentation (+ 8.39) shows the effectiveness of our method in increasing data informativeness. The network trained with

the final method only saw 300k data chunks during training, which is 14k and 249k less than the acquired only and simplified method respectively. Except for the other class, a significant increase in segmentation results is visible in all classes. The SP_{1_0} method has results similar to the simplified method and is superior to the direct uses of the segmentation network (acquired only) by 4.93 mIoU. The increased randomness in the data given to the network with the SP_{1_0} method increases its segmentation capability with the tank and the architectural classes at the cost of the metallic class (+10.2, +26.12 and -5.01 IoU respectively). Not reducing the dataset has dramatic effects on the segmentation results, with a loss of 7.66 mIoU. Those general results show the effectiveness of our method as well as the necessity of each of its components.

The improvements for the architectural class can be seen in the segmentation of the walls on the *piping* test scene (Fig. 6.2). Most of them are correctly segmented, even if a bleeding of this semantic class on the floor class is still visible in some cases (Fig. 6.3). A large part of the piping system is well understood, even with only acquired data. Pipes of medium bore size, which are the most represented objects in the training set, are quite well segmented in the *piping*(Fig. 6.3), *unit2*(Fig. 6.4) and *storage3*(Fig. 6.6) test scenes. However, the HVAC pipes in the *Plant* test scene are not understood without synthetic data (Fig. 6.7). There is no such example in the training dataset and the network is unable to generalise its knowledge, even with the SP_{1_0} method. This is consistent with findings in Section 4.2.1. Further improvements on the segmentation of those HVAC pipes when using the complete solution show the method ability to increase data informativeness.

The complete method also has the capacity to take notes of the less represented classes. Even if IoU for the electrical and other classes is still at the bottom end of the spectrum, the network tries to detect those classes. Errors in the *Unit2* test scene attest of this effort. On the other hand, when the dataset is not reduced, the increased data imbalance inhibit completely this effort to the point that this class is not even searched by the network (IoU of 0.00).

The increase in the segmentation quality of the tank class is more due to the reduction in error (the union part of IoU decreasing) than a better detection of tank object. The tank represented in the testing dataset is the most common example of storage tank (Fig. 6.6, Fig. 6.5) and is present in the acquired dataset. The quality of its segmentation does not greatly evolve between the three methods. Arguably, the results obtained by the SP_{1_0} method are slightly better than acquired only. However, the improvement on the piping and architectural classes reduced two of the biggest errors made by the network on the tank class. The last major error remaining is the understanding of a large piece of miscellaneous equipment as a tank in the *Plant* test scene (Fig. 6.7, Fig. 6.8). This mistake is perfectly understandable on the part of the segmentation network. This equipment look alike a storage tank and is not represented in either the acquired or synthetic data. Its presence in the testing data illustrate a difficulty of the task to be achieved in this thesis: some

types of equipment are uncommon and important but understandable in their context only. That the complete method segments this whole equipment as tank class is good enough. The project encompassing this thesis does not have the goal to perfectly segment everything. As long as segmentation errors are rare and coherent, they will be more easily manageable by users of the future solution based on this thesis. This is also why synthetic data are essential, due to its lack of data (even with increased informativeness), SP_{1_0} has a tendency to categorise unknown objects to the other class. This is the case with the electrical cables in the *Unit2* scene (Fig. 6.4), the wall in the *Piping* scene (Fig. 6.2) and a beam in the *Plant* scene (Fig. 6.8).

The increase of informativeness brought by the complete solution is generally linked with a more coherent segmentation. These are fewer points which are isolated from others of the same class. However, the current inference method is still a clear weak point. Lots of errors or discontinuities in semantic segmentation are due to this process. Some pillars are mislabelled by a lack of context or a change in point density (Fig. 6.5). Discontinuity occurs between neighbouring chunks (Fig. 6.3, Fig. 6.8).

This experiment validated the solution proposed in this thesis but also showed its limit. If overlapping data during training was proved effective (Sec. 5.1.2), a similar focus must be placed on the inference process. Before being able to improve further data informativeness during training, a work will have to be carried out on data understandability during inference.

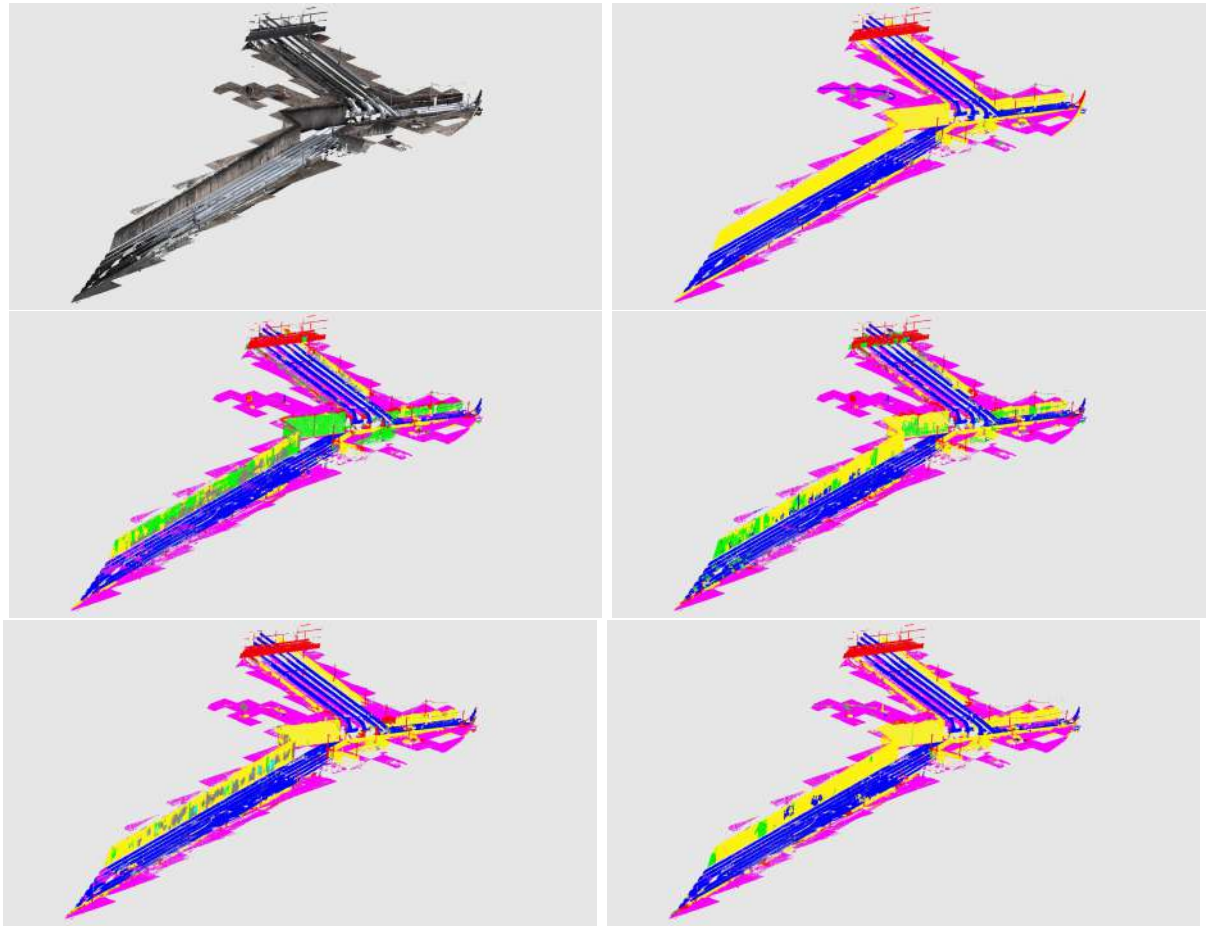


Figure 6.2: Inference result on the *Piping* test scene, front view. From left to right and top to bottom: raw data, ground truth, acquired data only, simplified method, SP_{1_0} method, complete method.

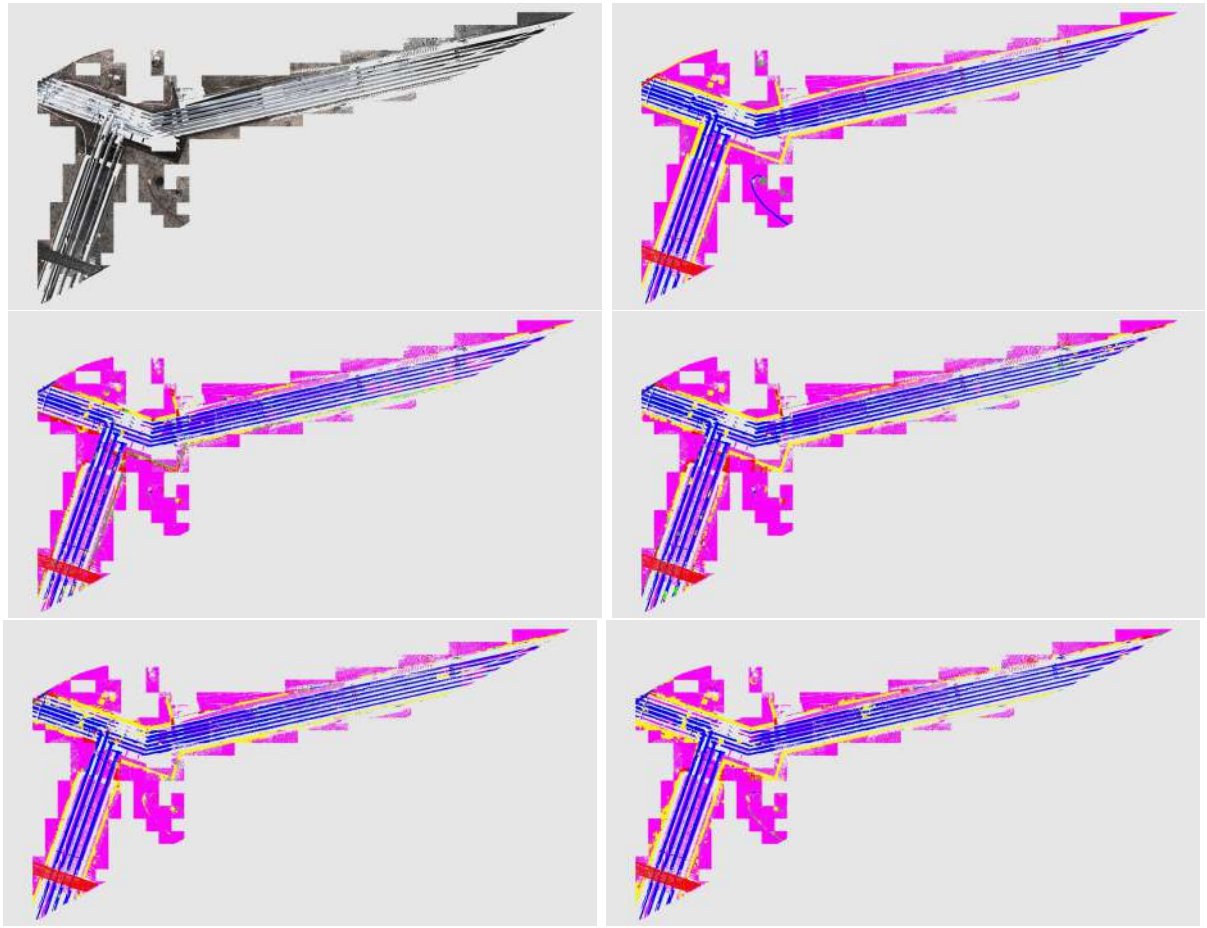


Figure 6.3: Inference result on the *Piping* test scene, top view. From left to right and top to bottom: raw data, ground truth, acquired data only, simplified method, SP_{1_0} method, complete method.

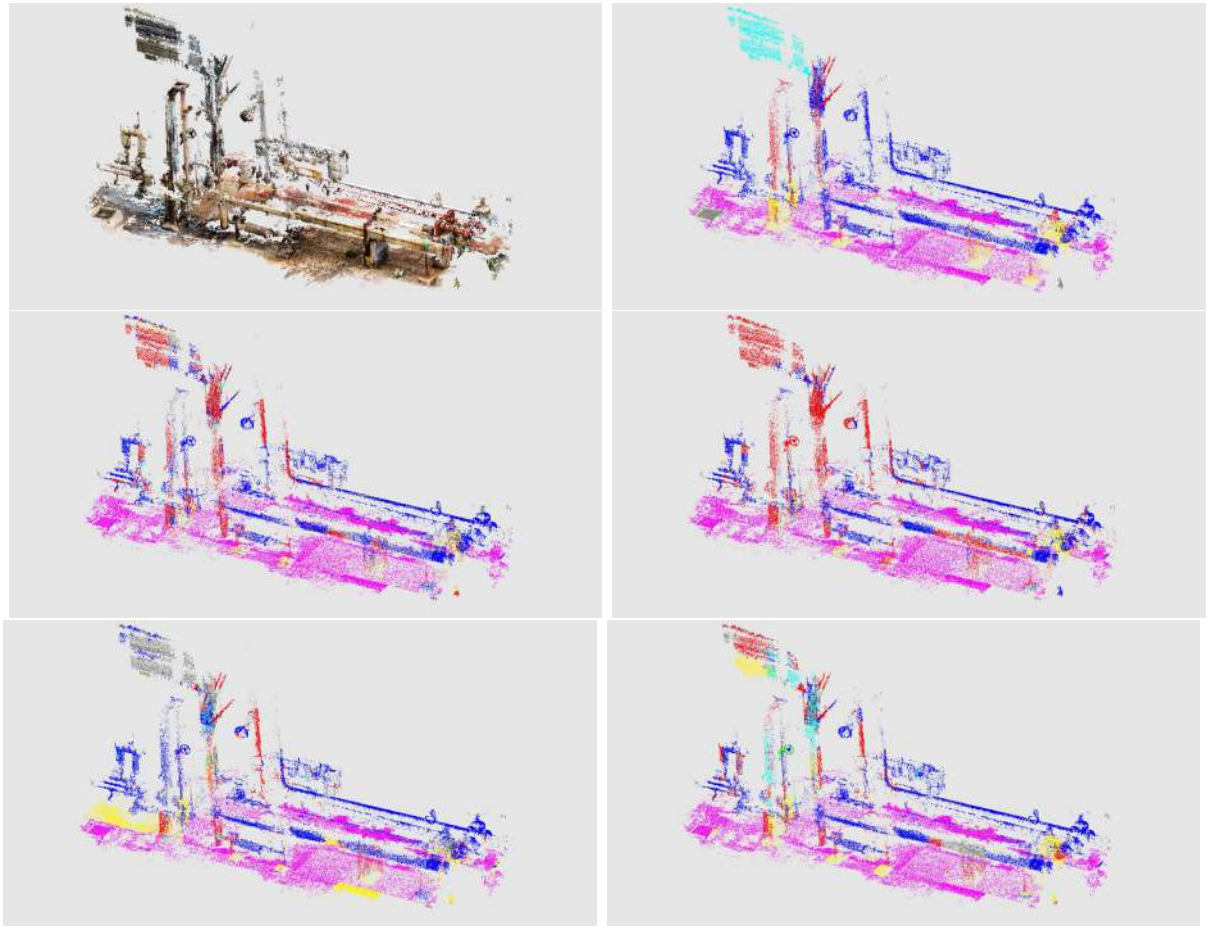


Figure 6.4: Inference result on the *Unit 2* test scene, front view. From left to right and top to bottom: raw data, ground truth, acquired data only, simplified method, SP_{1_0} method, complete method.

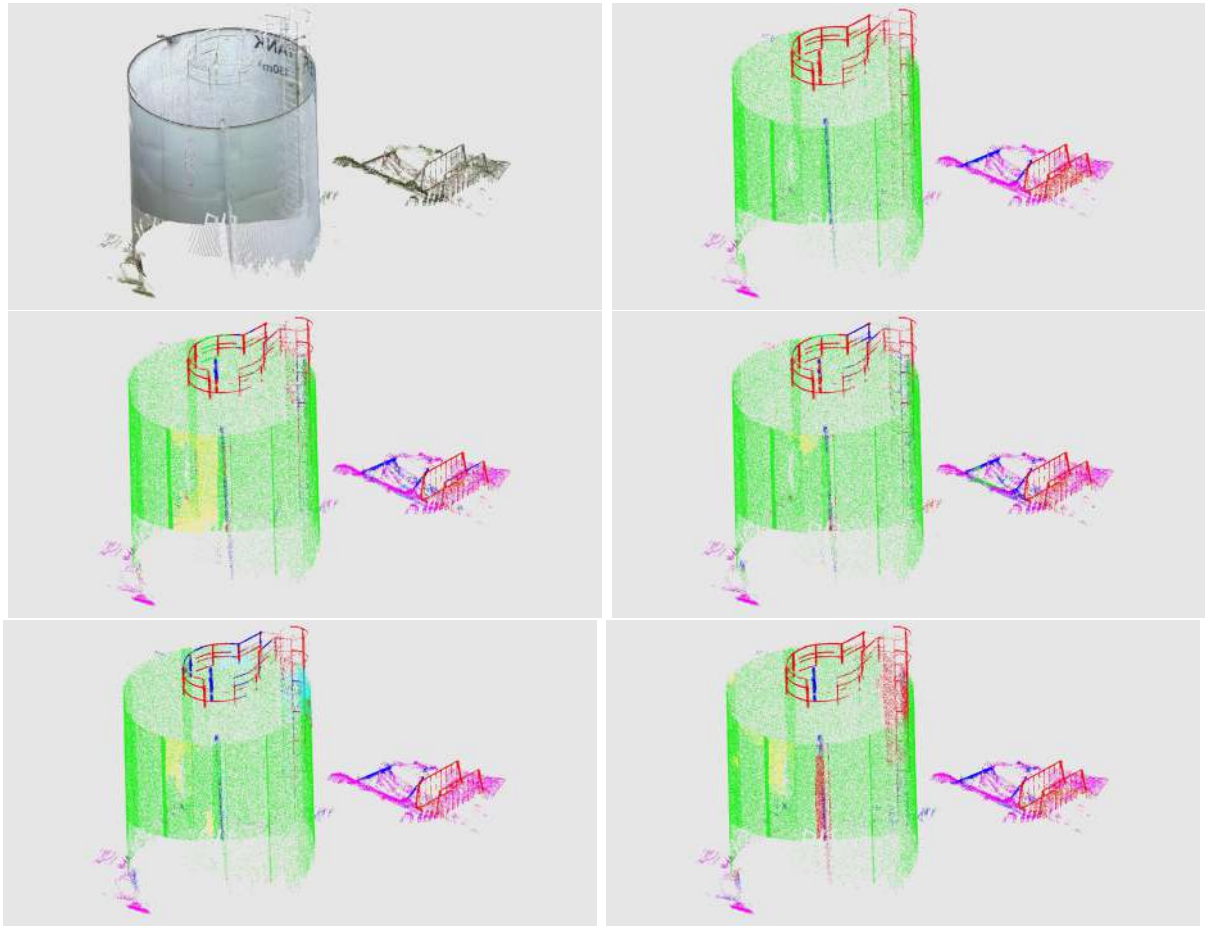


Figure 6.5: Inference result on the *Storage 3* test scene, back view. From left to right and top to bottom: raw data, ground truth, acquired data only, simplified method, SP_{1_0} method, complete method.

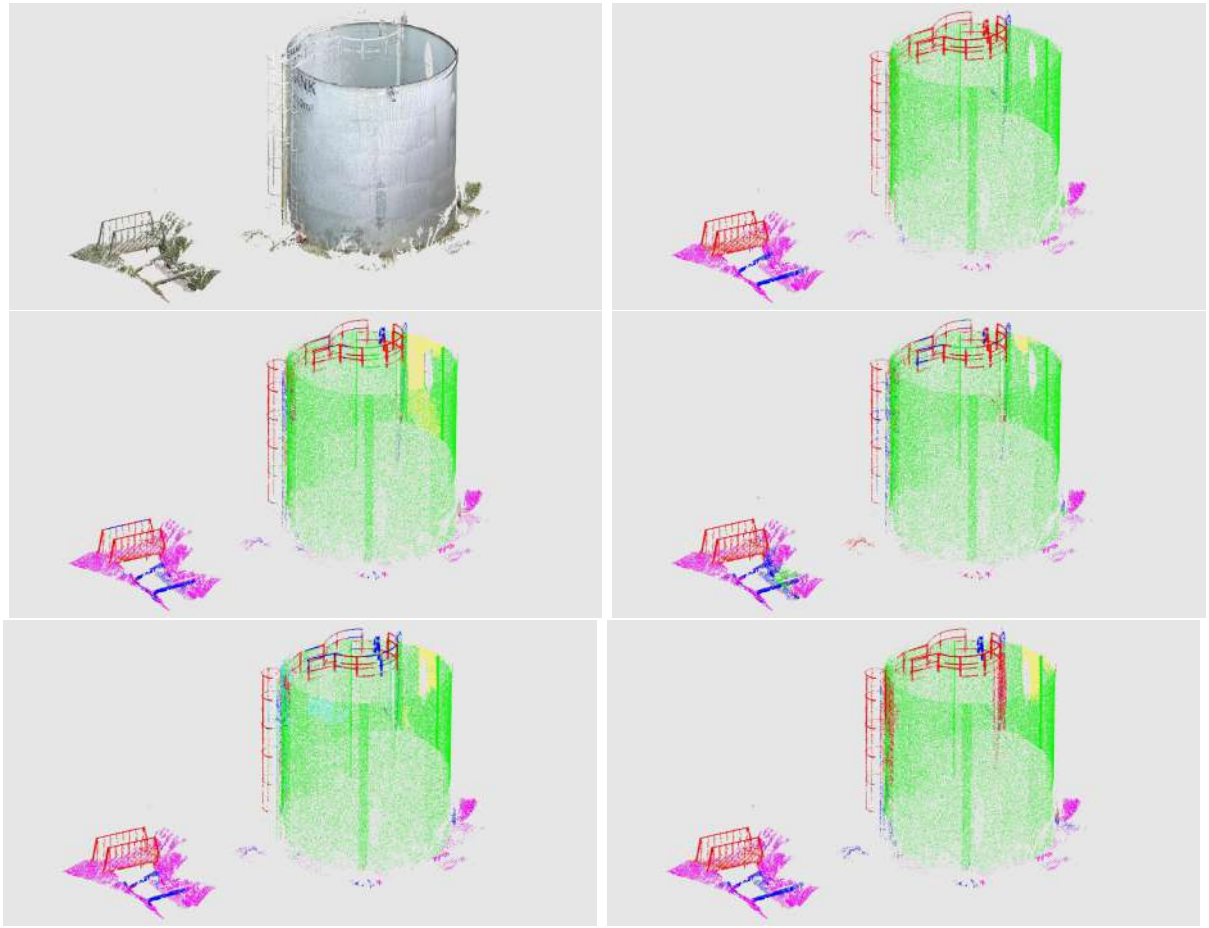


Figure 6.6: Inference result on the *Storage 3* test scene, front view. From left to right and top to bottom: raw data, ground truth, acquired data only, simplified method, SP_{1_0} method, complete method.

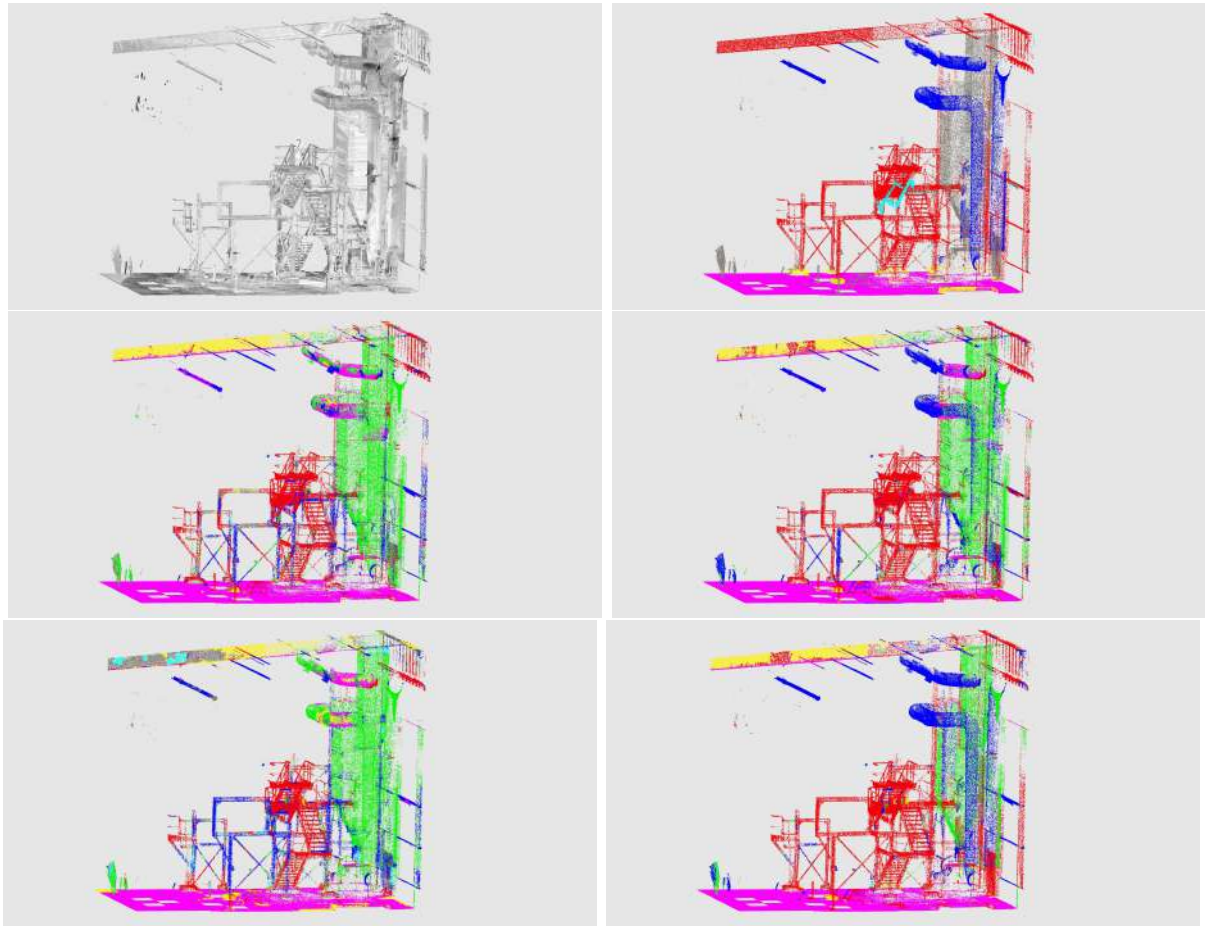


Figure 6.7: Inference result on the *Plant* test scene, front view. From left to right and top to bottom: raw data, ground truth, acquired data only, simplified method, SP_{1_0} method, complete method.

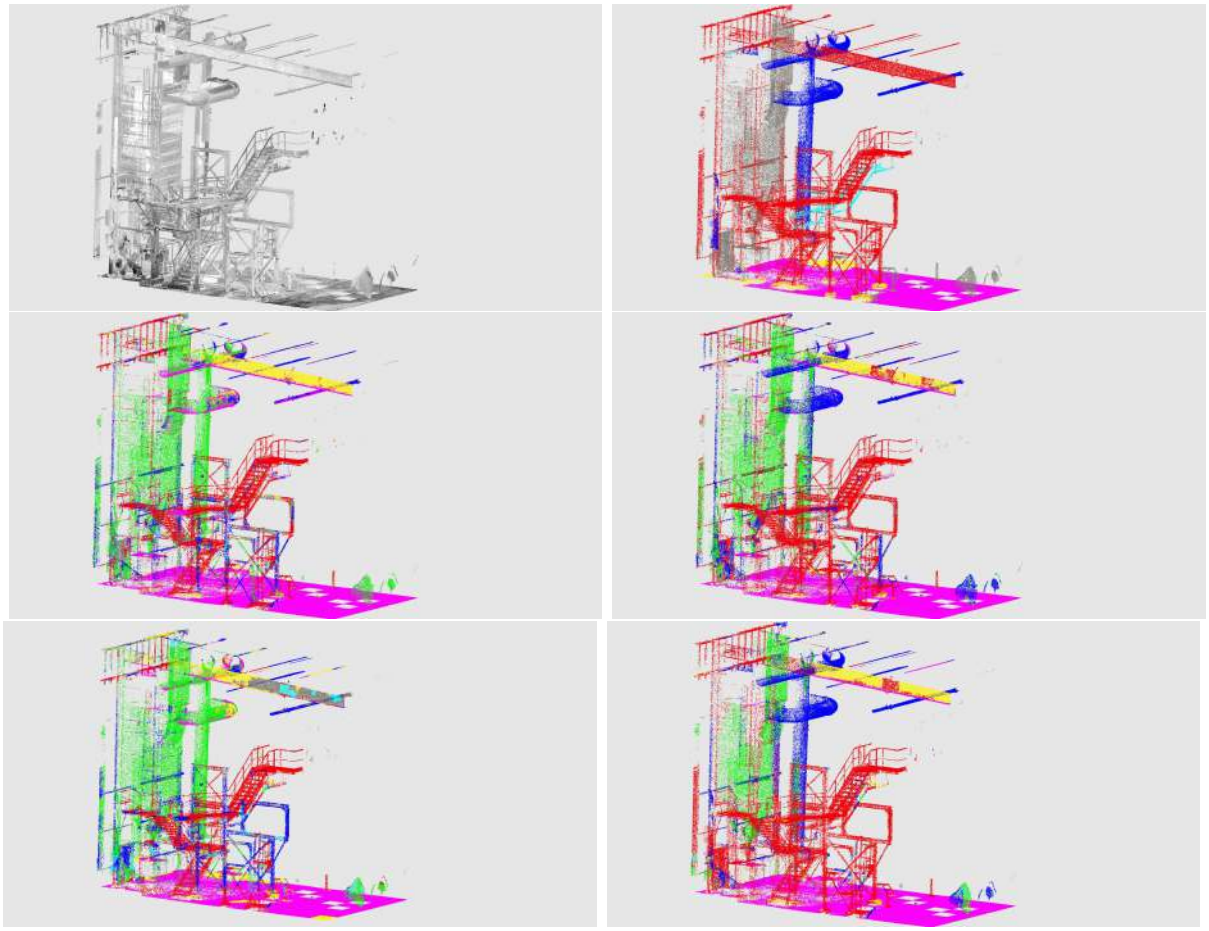


Figure 6.8: Inference result on the *Plant* test scene, back view. From left to right and top to bottom: raw data, ground truth, acquired data only, simplified method, SP_{1_0} method, complete method.

Chapter 7

Conclusion

Following the validation of the method developed during this thesis in Chapter 6, it is time to draw a conclusion on the work presented in this thesis by reviewing the answers it brings to the research questions (Sec. 7.1) but also its strength and weakness (Sec. 7.2). If, to our general research question (**GRQ**: **"Can point cloud semantic segmentation performance be increased by a more effective use of training data?"**), the answer is **"Yes, it can"**, there is still numerous ideas that can also be derived from our proposed solution (Sec. 7.3).

7.1 Conclusion

Chapter 2 contained an overview of the industrial methods for which point cloud semantic segmentation could be applied. The norm ISO-19650 was first described, before presenting recent works on deep learning applied to the industrial sector. The general problem of point cloud semantic segmentation was posed before exposing related works. This highlighted two gaps in the current literature: the lack of techniques to reduce domain shift for synthetic data augmentation when point clouds are concerned, as well as a lack of research on the influence of point cloud pre-processing towards the ability of deep neural networks to perform PCSS.

Following this, Chapter 3 laid the foundation for the work made in this thesis. It introduced a classification of industrial objects and determined which categories of objects are to prioritise for semantic segmentation. The peculiarities of data acquired from an industrial environment were also shown: the high variability in object relative size but also the scene layout, which differs greatly from the common applications of PCSS in the literature. Drawings from these factors, the general method explored in this thesis was introduced. It consists of a data processing framework divided in two parts. The first is specific to synthetic data and focuses on increasing their quality. The second part is applied to both acquired and synthetic data and tries to increase data informativeness by applying different pre-processing techniques.

The first experimental chapter, Chapter 4, detailed the synthetic data generation process devised in this thesis. It is able to work with scenes made of 3D meshes coming from a variety of origins (CAD models, manually made or automatically generated) and transform those meshes in point clouds. To reduce domain shift, this process uses a

Virtual Laser Sampling (VLS) method based on a novel virtual laser positioning method. A colouring method used to transfer colour information from unlabelled point clouds to colourless synthetic point clouds was also presented. Following the experiments carried out on this synthetic data generation process, synthetic data augmentation and VLS were validated on both a dataset of the literature, S3DIS[10], and our industrial dataset. The results on the colouring method were more nuanced. This method does improve semantic segmentation performance by colouring synthetic data, but only when the colouring of the whole point cloud data is similar. When data is acquired from different locations, the improvement brought forth by this method is marginal. The experiments answer RQ1: yes, synthetic data can be processed in a way to reduce domain gap.

Then, different concepts linked to point cloud data in general were studied in Chapter 5. Those studies allowed answering RQ2: the way data is presented to the semantic segmentation network does influence its performance. Strategies to divide point clouds in data chunks whose size is manageable by the semantic segmentation network were first explored. A strategy computing pillar shaped chunks overlapping with each other was selected. Following this exploration of data division, a strategy to improve the colouring method was subsequently devised. This strategy uses several colouring networks, each trained on data acquired from the same location, to create multiple versions of the same synthetic data coloured differently. A ratio parameter is defined to control the relative quantity of acquired and synthetic data seen by the segmentation network during training. Using a ratio of 0.5, segmentation mIoU increased by 0.51 compared to the best previous colouring method (22.97 to 23.48 mIoU). Data augmentations were studied in a second part. Computing point normal proved more successful as a point augmentation technique than computing point curvature. A random rotation around the vertical axis applied to each data chunk during training also proved effective to make the network robust towards rotation of data. Finally, the classification of data chosen initially was questioned and a simpler and more effective classification was determined. It groups objects belonging to the same system, such as piping or electrical, in a single class. This allows the network to focus less on small differences between highly represented classes with similar functions, such as structural and beams, and focus more on the less represented classes with completely different uses in the industrial domain, such as electrical components.

Finally, Chapter 6 concluded the experimental work made in this thesis by describing the retained solution following the tests made in previous chapters. The overall solution was tested by using a larger dataset than in the other parts. The performance attained, relative to the same semantic segmentation network using acquired data only, shows the effectiveness of the method developed in this thesis (+13.28 mIoU compared to initial results).

7.2 Discussion

Looking back at the work made towards presenting this thesis, some choices made can be discussed. Notably, the choice to work on a wide subject of research as well as the choice to keep the ability to compare most results during the thesis at a possible segmentation ability cost.

The first is the choice to make the research area quite wide but not particularly tall. A wide variety of methods were tested and experimented on. However, as will be seen in Section 7.3, most of these methods can be greatly extended. This is reflected in the general research question **(GRQ)**: **"Can point cloud semantic segmentation performance be increased by a more effective use of training data?"**. It also shows a choice made around the first year of work: building a foundation of knowledge to cover the hole in the literature, rather than trying to develop one highly performing state-of-the-art method. This differs from most of the current top publications in the domain which compete mostly on the few available datasets (Sec. 2.3.3.1). This choice makes it harder to directly and quantitatively compare the developed methods with other and future works should try to bridge this gap. It also influenced how **(RQ1)** (Can synthetic data be processed in a way that reduces domain gap?) and **(RQ2)** (Does the way data is presented to the segmentation network influence its performance?) were perceived during the works. The thesis is centred around the available data, acquired and synthetic, and was heavily influenced by them. Thus, even if the variety of methods developed in this thesis can be considered a success, the project could have been more focused which will have led to the creation of a more self-contained method. The foundation of knowledge developed is great but must be further developed and shared to attain its full potential. The research questions are all answered by an affirmative, but underlying questions remain.

Synthetic data can be processed in a way that reduces domain gap (RQ1). This was proved in Chapter 4 and by concurrent a work [103]. However, the experiments did not succeed in identifying all factors that allow creating informative synthetic data. The synthetic data used when segmenting data from the industrial domain also only originate from CAD models and manually made scenes with custom-made mesh models. Contrary to the works on S3DIS [10], the use of third party models was not explored. This is coherent with the original idea, to use CAD models as synthetic data, but does not explore the full possibilities of the method. The synthetic data used in most experiments, Synth3, is composed of two simple CAD models and a complex manually made model. It allowed the creation of interesting conditions even if using a complex CAD model would have been more in line with the original idea. The choice of experiments (the topology of synthetic data, or, what make a good synthetic data) is interesting but more practical questions could have also been explored. The necessary quantity of synthetic data to perform a good synthetic data augmentation would have been useful to tune the methods

using a ratio of acquired to synthetic data. Retrospectively, exploring more ways to generate synthetic data in order to balance the dataset could have been a better choice of study than dynamic balancing. However, this choice also implies reducing the thesis to RQ1 only.

The way data is presented to the segmentation network heavily influences its performance (RQ2). This question is particularly wide and focusing on either scene division or data augmentation could have led to more definite results. On the other hand, this wide point of view allowed us to solve one constraint defined by the overall project with the rotation augmentation (Sec. 5.2.2) and began to fill holes in the literature by working on scene division. Moreover, it led to the improvement of the colouring method and created a link between the two experimental parts of this thesis.

Lastly, two choices were made at the beginning of the work and not challenged in most of the thesis to ease comparison between results. They are the semantic segmentation network used and the dataset classification. A few tests were carried out with other network models (PointNet [75] and PointNet++ [76]) but they were far from extensive and lead to poorer results for a longer training time. For these reasons, they were not systematically carried out. However, validating some of the most important results in this thesis with another segmentation network would have allowed verifying their generality. Two methods where such tests would be the most beneficial are colouring and scene division. The dataset classification was decided from exchanges with industrial draughtsmen based on the data available at the beginning of the thesis. It created an interesting and realistic classification but which was not completely adapted to the method due to its imbalance. Retrospectively, this classification should have been changed to a hierarchical one, as in Section 5.3, at the end of the preliminary works and not when the whole data processing workflow was defined. The modification of the current segmentation network to one able to understand a hierarchy of labels is currently ongoing.

7.3 Future works

Following the work done in this thesis, several other ideas can be experimented, either on the scale of the thesis (PCSS) or by looking at deep learning on point cloud in general.

7.3.1 Synthetic data for sensor fusion

Sensor fusion based techniques are increasingly popular [36] and provide enhanced performance when both images and point clouds data are available (Sec. 2.3.3.2). However, the problem of lack of data is similar, if not worse, when considering sensor fusion. For each point cloud, corresponding images must be available. With a fully supervised method, both the images and the point cloud must be annotated. There is no such dataset repre-

senting industrial environments. Moreover, to the best of our knowledge, there is also no existing 2D image database of industrial facilities.

One approach [68] was successful in projecting labels created on CAD models to corresponding real images. Nevertheless, this approach only shifts the problem from needing corresponding point clouds and pictures to corresponding CAD models and pictures. If one is working with industrial draughtsmen, this kind of data can be more easily accessed by using past industrial projects. However, once this first hurdle is gone, three things are to be considered. Firstly, the viewpoints of the original images must be replaced in the corresponding CAD models. This work can be arduous, as the quality of the label projection will depend greatly on this step. Secondly, the camera model must be sufficiently well simulated to correctly project the labels. Errors around object edges will blur the segmentation network understanding in case of an improper image rendering. Lastly, each object in the image must be present in the CAD models, else miscellaneous elements will corrupt the network understanding of the different semantic classes. The difference between CAD models and pictured objects will also create mismatches in the dataset.

Using synthetic mesh data to augment a small, acquired, point cloud/picture fusion dataset seems to be a more cost-effective approach. This extends the work done in this thesis in several manners.

A work on linking meshes models to realistic texturing must be envisioned to create the images. The work made on virtual laser positioning must also be extended to virtual camera.

Being able to correctly apply texturing to 3D objects is not simple. The state-of-the-art method, ProcTHOR [31], uses predetermined textures on objects and only applies textures dynamically on surface objects (such as walls or floor).

The virtual laser positioning method was only tested with point cloud. However, a good viewpoint for positioning a laser should also be a good viewpoint for a camera. This assumption must be verified and the concept of density map extended correspondingly. Other considerations could appear when working with virtual camera. For example, the presence of walls or distance to objects. It is not problematic that some parts of the laser sampling acquire a wall. It is also not problematic if this wall is situated right next to the laser. Constraints exist when acquiring real point cloud and similar situations arise. However, on-site pictures do not contain only a wall or a close view of an object if there is no interesting information to take account of. Another set of parameters must be added to the problem when considering virtual camera positioning: the orientation of the camera. The findings that the laser model has next to zero influence on the segmentation results (Sec. 4.2.3) is also not guaranteed. This assumption could hold true if simple models of lenses are considered, such as kit lens or standard prime lens, which does not deform the image. When considering lenses which distort the images, such as fish-eyes lenses, the camera model definition will likely influence segmentation results. Depending on the case,

this could impede the project or offer an opportunity similar to the one explored for the colouration method (Sec. 5.1.4). Other camera parameters, such as focal length or focus, could also be considered.

When considering larger facilities, the laser positioning method also needs to be able to work on multiple floors without manual intervention. This can be easily done by transitioning to a 3D information density map. Such a map could reduce some problems associated to the current method:

- Tall object have a higher weight on the positioning algorithm decision than smaller ones. As the density map is constructed from a low-density Random Surface Sampling of the meshes, objects that are more tall than wide concentrate their points on few cells of the density map. Adding the vertical dimension to the density map would be fairer for long and wide objects.

- Visibility reduction is limited in a 2D density map. A beam three meters high has the same influence on visibility reduction than the same beam situated lower and in front of the laser. By using a 3D density map, the visibility reduction will offer more diverse viewpoints. However, in a large scene where there are wide empty zones, this could have an adverse effect on the segmentation by creating zones of low point-density. The relevance of a maximum range parameter for visibility reduction must be considered.

This transition from 2D to 3D is currently investigated. Slight changes in the algorithm must be done to counter the corresponding increase in computational power, similar to going from a pixel grid to a voxel grid. Using a definition of classes and surfaces where positioning lasers and cameras is possible seems to sufficiently reduce the algorithm computational cost by decreasing the number of potential viewpoint cells considered. Evaluating this 3D version of the positioning algorithm will first be done similarly than the experiments conducted in Section 4.2.3. Random, corner-only, 2D, 3D and 3D with visibility reduction positioning will all be considered. More complex CAD models will be used for this experiment. The envisioned models are a three-floor chemical processing unit and the 3 floors high, 140 x 70 meters, food processing plant used as example in the introduction.

7.3.2 Hybrid data

Mixing point cloud and synthetic data can also be considered. For example, the dataset proposed in PSNet5 [108] only contains a limited number of classes. Adding additional objects in this dataset, such as the floor or missing parts in the piping system can be done manually easily with a modelling software. With the appropriate 3D mesh models, an

annotated point cloud/mesh hybrid dataset can be constructed rapidly. Based on experience with modelling software, such as those used in Section 4.1.1 to generate synthetic data, converting PSNet5 [108] to such dataset can be done in one to two days of work.

However, to transform this hybrid dataset to a full point cloud dataset, two problems need to be studied: point cloud representation during VLS and laser positioning.

How to represent existing point clouds during VLS? They need to be accounted for during the computation of the density map but also to generate occlusion on the new objects. The simplest approach would be to use generic primitives such as cube and cylinder to represent these point clouds. In the case of PSNet5 [108], cube could represent beams, tanks and pumps, while cylinder would represent pipes. This method represents an easy to construct baseline. Following experiments on synthetic data topology (Sec. 4.2), this could prove sufficient. To go beyond this method, reconstructing the mesh from each point cloud object would be necessary. Preliminary tests show that the surface of such object is often irregular, bumpy and contains holes. The created meshes are also often left open due to occlusion during the corresponding point cloud acquisition. Tests comparing the baseline primitive method and the mesh reconstruction method would be necessary to determine if there is any need to research better reconstruction techniques. Several mesh reconstruction methods could be tested to precise this necessity.

The current laser positioning method may not be adapted to this case of hybrid point cloud/mesh data. Do the original laser positions need to be recovered and accounted for? The existing point clouds originate from an already done acquisition process and are subject to its constraints, the laser model and positions for example. Computing new laser positions could generate a discrepancy in the point cloud structure. This could in turn reduces informativeness of the point cloud for the segmentation network. The work carried out in Section 4.2 seems to hint at the fact that a discrepancy in local point structure will have almost no impact. However, it is inconclusive when larger structural defects, such as occlusion, are considered. Three possibilities exist for laser positioning:

- Recovering the initial laser positions. If they are provided, this step is straightforward. It can also be done manually when enough information is present in the point cloud. The floor class often shows discrepancy in its points-density around laser positions due to a laser minimum range. In case where such manual positioning is difficult, such as PSNet5 [108], a model could potentially be trained to accomplish such task. However, without the ability to look in both the fine-grained local-point-structure and the whole scene at the same time, which is an open research problem, this task could prove arduous to solve.

- Using new positions, computed via the current algorithm. With either a primitive-based or a reconstruction-based representation of the existing point clouds, this approach is straightforward.

- Using both initial and new positions. This approach presents the difficulties inherent to both of them but could allow multiple generations of synthetic data, thus increasing the total of available data.

The first approach is straightforward once the positions are recovered and the point clouds represented as meshes. This is not the case for the other two. They fail to correct the occlusion errors. The original point clouds will have occlusions that have no reason to exist but also points where occlusions should have occurred. Using the point cloud generated from VLS in lieu of the original point cloud would not make sense for this hybrid approach, as it represents too big a loss of data. Thus, works on both point removal to create occlusions but also mesh reconstruction to fill incorrect occlusions need to be done. As stated before, the current state of the art on point cloud reconstruction is quite lacking. Thus, this last two laser positioning methods could prove to be a difficult to use fully.

Finally, the colouring of the synthetic data must be coherent with the original point cloud. It is not evident if the current method would be able to create a coherent colouring. Training on the original cloud will not teach the colour features of the missing objects (those to be coloured at inference) to the colouring network. Training on other datasets will generate a discrepancy in the colours of the final point cloud. Moreover, as the method struggles when the data used to train the colouring network originate from multiple locations (Sec. 4.3.1), it is not possible to mix the original points and another dataset. Using an adversarial approach seems to be necessary to solve this problem and improve the colouring method. A patch-based approach, mixed with deep learning, could also be used, as done in works in grey-scale image colouring or image denoising [105].

7.3.3 Augmenting the framework

The current framework used for the data processing workflow is quite flexible. Other methods could easily be plugged in, either for increasing synthetic data quality or augmenting data as a whole.

Combining our method with other working on different sources of domain shift in the data can be considered. For example, the work made on SynLiDAR [103], considers both appearance and sparsity of point cloud. Multiplying the kind of lasers used, similar to the method used with colouring (Sec. 5.1.4), could also prove beneficial. Even if the differences are minimal between laser models, it is similar to adding noise to data during training, a common trick in machine learning. As we are using Helios++ [99], it is possible to define a material for each mesh. This property will influence the results of VLS. If the deformation created by the material is large enough, this provides another opportunity to decrease domain shift or diversify available data. The material also influences the intensity field of the point cloud. It is currently not used as point cloud originating from

photogrammetry does not possess such value. Using it as an additional feature, as with colour and normals, could be an interesting static augmentation. Two choices are to be studied when the information is unavailable: using a default value or computing a value with a model similar to SqueezeSeg V2 [101].

More than intensity, additional data augmentations can be considered such as symmetry and colour noise. For symmetry, following the results on the rotation augmentation (Sec. 5.2.2), it is preferable to only use those that keep the vertical orientation of the data chunk. As colour is an important feature which the network rely heavily on (Sec. 4.3.1), an idea similar to dropout [86] could be applied. The colour value of a chunk as a small chance to be erased during training. This is sometimes known as colour annealing [91]. The effect of this augmentation could be further researched, for example by not simply erasing colour but applying a uniform or a random colouring. Concerning static augmentation, adding curvature information to the data failed, however, other features could be used to augment data, such as the planarity or sphericity of the points, as defined in [42].

Finally, following our choice to only work on data without modifying the network, it should not have adverse effects on existing methods that modify the segmentation network in order to cope with domain shift. Thus, combination of our method with those should be considered. However, if the domain shift is lower, do they still bring an advantage to the network during training?

7.3.4 PCSS network input

When considering the application of deep learning techniques to point cloud in a wider sense, working with large point cloud scenes is an ongoing problem (Sec. 2.3.3.3). Two categories of solutions are possible against this problem: dividing the scene into smaller chunks or trying to work on the whole scene.

The segmentation method used in this thesis is of the first category. Our works on finding an appropriate chunk division for our industrial dataset and the S3DIS dataset [10] showed that, depending on the data, the chunk shape should be changed to improve segmentation results. As seen in Chapter 5, S3DIS reacted more favourably to an increase in chunk size than SMARI, even if the best chunk shape found is a 1 x1 meter pillar for both. This optimal size seems to depend both on point-density and the shape and size of the objects to be segmented. Extending this work to other datasets, which offers different conditions of object shape and point-density, could help in defining guidelines for point cloud division. The SemanticKITTI [14] is considered as it seems sparser than S3DIS but also contains larger objects. A complete analysis of the dataset, similar to the one carried out in Section 3.2 still need to be done to confirm this fact. Another aspect to consider is the influence of the segmentation method towards chunk shape. Methods working on complete scenes or large chunks, such as RandLA-Net [46], use

specific layers. The network, KP-FCNN [92], used in the thesis is convolution-based. MLP based methods possess their own strengths and weaknesses [40], [76]. Thus, the influence of the segmentation network towards the optimal chunk shape could be important and must be estimated in order to create a guideline for point cloud division.

Dividing the scene into chunks can either be done in a static or a dynamic manner. The advantage of the static manner is that it offers more control on the chunk contents. For now, it is the one which works the best. It is possible to balance a dataset by working on those chunks [39]. The work in Section 5.1.3 tried to replicate this idea dynamically. The current method fails to beat the static method. Following the last observations made in Section 5.1.3, a few ideas can be proposed. Firstly, by defining borders inside the pre-computed chunks to ensure the creation of appropriately sized chunks in most cases. The necessity to randomly create smaller sized chunks must also be considered in order to bring more diversity to the data observed by the network. The creation of those smaller chunks could be accompanied by a drop in point-density. This would replicate what happens at the border of a test scene during inference. Secondly, other metrics than the allocation vector A could be used. A process closer to the work of Griffiths et al. [39], where the number of augmentation for each chunk is precomputed, could be envisioned. Lastly, precomputing metrics on the whole dataset before training could help in choosing chunks dynamically. The uses of those metrics could have an adversarial effect, similar to the number strategy (Sec. 5.1.3). To alleviate this possible problem, combining this approach with a allocation vector A should also be studied.

Those two works are currently being investigated and are considered for publication if they prove to be successful or insightful enough.

7.3.5 Considering output

The work presented in this thesis focuses exclusively on data and every method that can employed to increase their effectiveness before the input layer of the segmentation network. Chapter 6 showed the improvement brought by such method but also its limitations. Due to a lack of context during inference, some objects are badly understood. For example, the distinction between a tank and a wall is not always correctly made. Linking some findings on scene division with the output of the network could improve segmentation results.

The work carried out in KPConv [92] can be taken as an example. In the proposed inference process, the scene is sampled by spheres that are partially randomly drawn. This drawing process is led by a point potential computation which ensures that most points in the scene are seen the same amount of time. This strategy significantly improves segmentation results but is computationally costly. It was not used in this thesis for this reason, as it violates a constraint brought by the general project (Sec. 3.4.2). A first

way to improve inference results would then be to improve this process by reducing its computational cost. The potential solution ranges from simple parametric ones, such as decreasing the potential to be reached by each point, to modifying the complete potential mechanism.

Looking at the failure case illustrated in Chapter 6, sharing the information of each chunk to their neighbourhood can be contemplated. This could be done in a two steps process. The first step is similar to the one used in the thesis, a network segment each chunk independently. The second step then propagates the segmentation information to neighbours. A first way to propagate this information is by using a recurrent network [106][47]. In this case, extracting features from each chunk segmentation results is left to the network. A possible difficulty is propagating information in every direction as the order in which chunks are fed to a recurrent network influence its decision. A graph-oriented approach could also be used, either by considering chunks as superpoints [55] or encoding the chunks with a Variational Auto-Encoder, similarly to VV-Net [66] but by replacing small voxels by meters sized chunks. One advantage of these information-sharing strategies is their potential ability to work with ball-shaped chunks, such as cube. The transfer of information in a pillar is limited by the number of layer in the network and the subsampling process. In an extremely vertical scene, such as the *plant* test scene, large beams near the ceiling are confused with a floor plane with structural or architectural elements on top. Using ball-shaped chunks optimally sized for the network would ensure a good sharing of context knowledge in the deepest layer of the network. The envisioned inter-chunk knowledge transfer strategies would greatly alleviate the weakness of such chunks (Sec. 5.1.2).

Appendix A

Publications

- **R. Cazorla**, L. Poinel, P. Papadakis, and C. Buche. "Bottleneck Identification to Semantic Segmentation of Industrial 3D Point Cloud Scene via Deep Learning". In: *Thirtieth International Joint Conference on Artificial Intelligence*, Vol. 5, Aug. 2021, pp. 4877- 4878, A* ERA CORE rank. DOI: 10.24963/ijcai.2021/670.
- **R. Cazorla**, L. Poinel, P. Papadakis, and C. Buche. "Reducing Domain Shift in Synthetic Data Augmentation for Semantic Segmentation of 3D Point Clouds". In: *IEEE Conference on Systems, Man, and Cybernetics 2022*, Prague, Oct. 2022, pp. 1190-1197, B ERA CORE rank. DOI: 10.1109/SMC53654.2022.9945480.
- **R. Cazorla**, L. Poinel, P. Papadakis, and C. Buche. "Enhancing synthetic data generation for semantic segmentation of point clouds". Accepted at: *SPIE Defense + Commercial Sensing, Geospatial Informatics XIII*, Orlando, May 2023.

Appendix B

Additional results

Paper	Dataset	Data used	# points training	# points testing	Data division	Augmentation
PointNet [75]	S3DIS	XYZ, RGB, normalised location	4096	All	Room/1×1 m pillar	-
[34]	S3DIS	XYZ, RGB, normalised location	4096	4096	Room/1×1 m pillar	-
DGCNN [98]	S3DIS	XYZ, RGB, normalised location	4096	-	Room/1×1 m pillar	-
GACNet [93]	S3DIS	-	4096	All	Room/1×1 m pillar	-
LSP [55]	S3DIS	-	-	-	Scene	-
3D-RNN [106]	S3DIS	Normalised XYZ, RGB	6400	-	Room/1.5×1.5 m pillar	-
A-CNN [52]	S3DIS	Normalised XYZ, RGB	4096	-	Room/1×1 m pillar	Pillars scaled to a height of 2 m. Point permutation.
PointSIFT [49]	S3DIS	-	-	-	Room/1×1 m pillar	-
PointWeb [109]	S3DIS	XYZ, RGB, normalised location	4096	All	Room/1×1 m pillar	-
PointWeb [109]	S3DIS	XYZ	8192	All	1.5×1.5 m pillar	-
SGPN [95]	S3DIS	XYZ, RGB, normalised location	4096	All	Room/1×1 m pillar	-
JSIS3D [73]	S3DIS	XYZ, RGB	4096	4096	-	-
ASIS [96]	S3DIS	XYZ, RGB, normalised location	4096	All	Room/1×1 m pillar	-
PointCNN [58]	S3DIS	Normalised XYZ, RGB	-	-	Room/1.5×1.5 m pillar	Rotation
RSNet [47]	S3DIS	XYZ, RGB, normalised location	4096	-	Room/1×1 m pillar	-
RandLA-Net [46]	S3DIS	XYZ, RGB	10^5	All	Room	-
KPConv [92]	S3DIS	XYZ, RGB	-	-	2 m radius sphere	-
MSSCN [33]	S3DIS	XYZ, RGB	16384	-	-	-
SAN [18]	S3DIS	XYZ	4096	-	-	-
FPVC [97]	S3DIS	XYZ, RGB	8192	All	Room/2×2 m pillar	-

Table B.1: Overview of the input format of point-based PCSS methods available in the literature when the S3DIS [10] dataset is used.

Paper	Dataset	Data used	# points training	# points testing	Data division	Augmentation
PointNet++ [76]	ScanNet	XYZ	8192	-	1.5×1.5×3 m block	-
3D-RNN [106]	ScanNet	XYZ	6400	-	Room/1.5×1.5 m pillar	-
A-CNN [52]	ScanNet	Normalised XYZ, Normals	8192	-	Room/1.5×1.5 m pillar	Pillars scaled to a height of 2 m. Point permutation.
PointSIFT [49]	ScanNet	-	-	-	-	-
PointCNN [58]	ScanNet	Normalised XYZ	-	-	Room/1.5×1.5 m pillar	Rotation
RSNet [47]	ScanNet	XYZ	4096	-	1×1 m pillar	-
PointCONV [102]	ScanNet	XYZ, RGB	8192	-	1.5×1.5×3 m block	-
MV-PointNet [48]	ScanNet	-	8192	-	1.5×1.5 m pillar	Rotation
UPBF [28]	ScanNet	-	8192	-	1.5×1.5 m pillar	Rotation
MSSCN [33]	ScanNet	XYZ	16384	-	-	-
SAN [18]	ScanNet	-	8192	-	-	-
LatticeNet [8]	ScanNet	-	-	-	Scene	-
GACNet [93]	Semantic3D	-	4096	All	4×4 m pillar	-
RandLA-Net [46]	Semantic3D	XYZ, RGB	10^5	All	-	-
$(AF)^2$-S3NET [27]	SemanticKITTI	-	-	-	-	-
LatticeNet [8]	SemanticKITTI	-	-	-	Scan	-

Table B.2: Overview of the input format of point-based PCSS methods available in the literature with the scanNet [30], Semantic3D [41] and SemanticKITTI [14] datasets.

Method	Acc	mIoU	Pipe	Beam	Valve	Tank	Floor	Walkway	Struct.
Acquired Only	69.30	18.82	66.67	0.27	**5.76**	32.29	82.55	42.42	28.11
Simplified	73.64	22.97	72.61	29.02	5.54	41.51	82.94	50.27	38.07
Complete	**81.51**	**27.69**	**82.70**	**34.34**	2.19	**59.10**	**86.23**	**58.80**	**59.80**

Table B.3: Results obtained by the method devised during this thesis, using the initial classification.

Appendix C

Industrial objects

Hereafter are illustrated some common industrial objects.

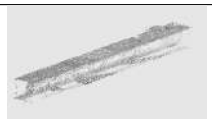
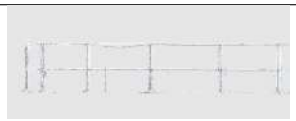
Object type category	Examples
Structural and mettalic objects	
T-brace	
Wide flange	
Circular hollow	
Handrails	
T shape	
I-beam	
Walkway	
Architectural objects	
Wall panel	
Foundation	

Table C.1: Example and illustration of most common industrial objects.

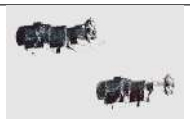
Object type category	Examples
Piping objects	
Tee	
Elbow	
Valve	
Flange	
Reducer	
Pipe	
Heating, Ventilation and Air Cooling (HVAC)	
Equipment	
Vessel	
Pump	
Tank	

Table C.2: Example and illustration of most common industrial objects.

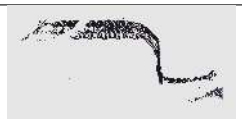
Object type category	Examples
Electrical objects	
Cable tray	
Conduit	
Electrical panel	
Lights	
Instrumentation	
Barometer	
Safety objects	
Fire extinguisher	

Table C.3: Example and illustration of most common industrial objects.

Appendix D

Tools

D.1 Software used

Hereafter is included a list of software used to generate the results presented in this manuscript.

- CloudCompare 2.12
- Blender 2.93.6
- AutoCAD 2020
- Rhinoceros 7
- Visual Studio 2019
- PyCharm Community Edition 2020.3.5
- Anaconda 3
- Git 2.33.0

D.2 Programming languages used

The C# code used to pre-process data and extracts results is based on .Net Framework 4.7.1 with no external dependencies. The C++ scene generator used C++ 14, with the irrlicht-1.8.4, eigen-3.4.0 and chrono (commit version b5af3f) libraries.

The following python libraries were used in the developing machine for developing, training and testing the different neural networks and data processing methods. Almost the same set of library was used for the two computing machine. Due to the difference in hardware, variations are on the CUDA distribution and the python libraries using the GPU, such as torch.

Package	Version	Package	Version	Package	Version
absl-py	0.14.0	jupyter-client	7.0.3	pywinpty	1.1.4
addict	2.4.0	jupyter-core	4.8.1	PyYAML	5.4.1
antlr4-python3-runtime	4.8	jupyterlab-pygments	0.1.2	pyzmq	22.3.0
argcomplete	1.12.3	jupyterlab-widgets	1.0.2	rdflib	6.0.1
argon2-cffi	21.1.0	kiwisolver	1.3.2	requests	2.26.0
ase	3.22.1	llvmlite	0.33.0	requests-oauthlib	1.3.0
attrs	21.2.0	Markdown	3.3.4	rsa	4.7.2
backcall	0.2.0	MarkupSafe	2.0.1	scikit-image	0.16.2
bleach	4.1.0	matplotlib	3.4.3	scikit-learn	1.0
cached-property	1.5.2	matplotlib-inline	0.1.3	scipy	1.6.1
cachetools	4.2.2	mistune	0.8.4	seaborn	0.12.2
certifi	2021.5.30	nbclient	0.5.4	Send2Trash	1.8.0
cffi	1.15.0	nbconvert	6.2.0	sentry-sdk	1.4.1
charset-normalizer	2.0.6	nbformat	5.1.3	setuptools	41.2.0
click	8.0.1	nest-asyncio	1.5.1	shortuuid	1.0.1
colorama	0.4.4	networkx	2.6.3	six	1.16.0
configparser	5.0.2	ninja	1.11.1	sklearn	0.0
cycler	0.10.0	notebook	6.4.4	smmap	4.0.0
debugpy	1.4.3	numba	0.50.1	subprocess32	3.5.4
decorator	5.1.0	numpy	1.19.5	tensorboard	2.6.0
defusedxml	0.7.1	nvidia-ml-py3	7.352.0	tensorboard-data-server	0.6.1
docker-pycreds	0.4.0	oauthlib	3.1.1	tensorboard-plugin-wit	1.8.0
entrypoints	0.3	omegaconf	1.4.1	terminado	0.12.1
filelock	3.1.0	open3d	0.12.0	testpath	0.5.0
gdown	3.13.1	packaging	21.0	threadpoolctl	2.2.0
gitdb	4.0.7	pandas	1.1.5	torch	1.8.1+cu102
GitPython	3.1.24	pandocfilters	1.5.0	torch-cluster	1.5.9
google-auth	1.35.0	parso	0.8.2	torch-geometric	1.6.3
google-auth-oauthlib	0.4.6	pickleshare	0.7.5	torch-scatter	2.0.9
googledrivedownloader	0.4	Pillow	8.3.2	torch-sparse	0.6.12
gql	0.2.0	pip	19.2.3	torchfile	0.1.0
graphql-core	1.1	plyfile	0.7.4	torchnet	0.0.4
graphviz	0.19.1	prometheus-client	0.11.0	torchvision	0.9.1+cu102
grpcio	1.40.0	promise	2.3	torchviz	0.0.2
h5py	3.4.0	prompt-toolkit	3.0.20	tornado	6.1
hydra-core	0.11.3	protobuf	3.18.0	tqdm	4.62.3
idna	3.2	psutil	5.8.0	traitlets	5.1.0
imageio	2.9.0	pyasn1	0.4.8	types-requests	0.1.13
importlib-metadata	4.8.1	pyasn1-modules	0.2.8	types-six	0.1.9
importlib-resources	5.2.2	pybind11	2.8.1	typing-extensions	3.10.0.2
install	1.3.5	pycparser	2.20	urllib3	1.26.7
ipykernel	6.4.1	Pygments	2.10.0	visdom	0.1.8.9
ipython	7.28.0	pyparsing	2.4.7	wandb	0.8.36
ipython-genutils	0.2.0	pyquaternion	0.9.9	watchdog	2.1.5
ipywidgets	7.6.5	pyrsistent	0.18.0	wcwidth	0.2.5
isodate	0.6.0	PySocks	1.7.1	webencodings	0.5.1
jedi	0.18.0	python-dateutil	2.8.2	websocket-client	1.2.1
Jinja2	3.0.1	python-louvain	0.15	Werkzeug	2.0.1
joblib	1.0.1	pytorch-metric-learning	0.9.99	wheel	0.37.0
jsonpatch	1.32	pytz	2021.1	widgetsnbextension	3.5.1
jsonpointer	2.1	PyWavelets	1.1.1	zipp	3.5.0
jsonschema	3.2.0	pywin32	301		

BIBLIOGRAPHY

[1] AFNOR, *NF EN ISO 19650-1*, Dec. 2018 (cit. on pp. viii, 10).

[2] AFNOR, *NF EN ISO 19650-2*, Dec. 2018 (cit. on p. 10).

[3] AFNOR, *NF EN ISO 19650-3*, Aug. 2020 (cit. on p. 10).

[4] AFNOR, *NF EN ISO 19650-4*, Sept. 2022 (cit. on p. 10).

[5] AFNOR, *NF EN ISO 19650-5*, July 2020 (cit. on p. 10).

[6] Evangelia Agapaki, "Automated Object Segmentation in Existing Industrial Facilities", Thesis, University of Cambridge, July 18, 2020, DOI: 10.17863/CAM.52102 (cit. on pp. 1, 2, 12, 32, 38–40, 56).

[7] article Evangelia Agapaki, Alex Glyn-Davies, Sara Mandoki, and Ioannis Brilakis, "CLOI: A Shape Classification Benchmark Dataset for Industrial Facilities", *in*: *ASCE International Conference on Computing in Civil Engineering*, 2019, pp. 66–73, DOI: 10.17863/CAM.36600 (cit. on pp. 38, 40, 41).

[8] article Radu Alexandru Rosu, Peer Schütt, Jan Quenzel, and Sven Behnke, "LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices", *in*: *Robotics: Science and Systems XVI*, July 12, 2020, DOI: 10.15607/RSS.2020.XVI.006 (cit. on pp. 19, 23, 24, IV).

[9] Iro Armeni, Sasha Sax, Amir R. Zamir, and Silvio Savarese, "Joint 2D-3D-Semantic Data for Indoor Scene Understanding", Apr. 5, 2017, arXiv: 1702.01105 [cs] (cit. on pp. 21, 35, 36, 40, 45, 67, 68, 73).

[10] article Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese, "3D Semantic Parsing of Large-Scale Indoor Spaces", *in*: *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1534–1543 (cit. on pp. 21, 23, 26, 47, 94, 103, 106, 108, 113, 122, 138, 139, 145, III).

[11] online Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**.*12* (Dec. 1, 2017), pp. 2481–2495, DOI: 10.1109/TPAMI.2016.2644615 (cit. on p. 20).

[12] Jim Bedrick, Will Ikerd, and Jan Reinhardt, *Level Of Development (LOD) Specification Part 1*, BIM Forum, 2021 (cit. on pp. 32, 33).

[13] online Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Jürgen Gall, and Cyrill Stachniss, "Towards 3D LiDAR-based Semantic Scene Understanding of 3D Point Cloud Sequences: The SemanticKITTI Dataset", *in*: *The International Journal of Robotics Research* **40**.*8-9* (Aug. 2021), pp. 959–967, DOI: 10.1177/02783649211006735 (cit. on p. 22).

[14] article Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences", *in*: *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 9297–9307 (cit. on pp. 6, 22, 26, 35, 53, 123, 145, IV).

[15] online Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert, "SnapNet: 3D Point Cloud Semantic Labeling with 2D Deep Segmentation Networks", *in*: *Computers and Graphics* **71** (2018), pp. 189–198, DOI: 10.1016/j.cag.2017.11.010 (cit. on pp. 18, 24).

[16] article Aurelie Bugeau and Vinh-Thong Ta, "Patch-Based Image Colorization", *in*: *21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 3058–3061 (cit. on p. 55).

[17] article Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving", *in*: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 11618–11628, DOI: 10.1109/CVPR42600.2020.01164 (cit. on pp. 22, 35).

[18] online Guorong Cai, Zuning Jiang, Zongyue Wang, Shangfeng Huang, Kai Chen, Xuyang Ge, and Yundong Wu, "Spatial Aggregation Net: Point Cloud Semantic Segmentation Based on Multi-Directional Convolution", *in*: *Sensors* **19**.*19* (19 Jan. 2019), p. 4329, DOI: 10.3390/s19194329 (cit. on pp. 24, III, IV).

[19] article Xu Cao and Katashi Nagao, "Point Cloud Colorization Based on Densely Annotated 3D Shape Dataset", *in*: *Lecture Notes in Computer Science*, vol. 11295, 2019, pp. 436–446, DOI: 10.1007/978-3-030-05710-7_36 (cit. on p. 28).

[20] article Xu Cao, Weimin Wang, Katashi Nagao, and Ryosuke Nakamura, "PSNet: A Style Transfer Network for Point Cloud Stylization on Geometry and Color", *in*: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2020, pp. 3326–3334, DOI: 10.1109/WACV45572.2020.9093513 (cit. on p. 28).

[21] online F. M. Carlucci, P. Russo, and B. Caputo, "(DE)$\hat{2}$ CO: Deep Depth Colorization", *in*: *IEEE Robotics and Automation Letters* **3**.*3* (July 2018), pp. 2386–2393, DOI: 10.1109/LRA.2018.2812225 (cit. on pp. 28, 29).

[22] article Romain Cazorla, Line Poinel, Panagiotis Papadakis, and Cédric Buche, "Bottleneck Identification to Semantic Segmentation of Industrial 3D Point Cloud Scene via Deep Learning", *in*: *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 5, Aug. 9, 2021, pp. 4877–4878, DOI: 10.24963/ijcai.2021/670 (cit. on p. 34).

[23] article Romain Cazorla, Line Poinel, Panagiotis Papadakis, and Cedric Buche, "Reducing Domain Shift in Synthetic Data Augmentation for Semantic Segmentation of 3D Point Clouds", *in*: *IEEE Systems, Man and Cybernetics (SMC)*, Oct. 10, 2022, pp. 1190–1197, DOI: 10.1109/SMC53654.2022.9945480 (cit. on pp. 48, 53, 67, 73).

[24] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu, *ShapeNet: An Information-Rich 3D Model Repository*, Stanford University, Princeton University, Toyota Technological Institute at Chicago, Dec. 9, 2015, arXiv: 1512.03012 (cit. on p. 67).

[25] article Thomas Chaton, Nicolas Chaulet, Sofiane Horache, and Loic Landrieu, "Torch-Points3D: A Modular Multi-Task Framework for Reproducible Deep Learning on 3D Point Clouds", *in*: *International Conference on 3D Vision (3DV)*, Nov. 2020 (cit. on p. 45).

[26] article Liang Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation", *in*: *European Conference on Computer Vision (EECV)*, vol. 11211 LNCS, Feb. 7, 2018, pp. 833–851, DOI: 10.1007/978-3-030-01234-2_49 (cit. on p. 20).

[27] article Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu, "(AF)2-S3Net: Attentive Feature Fusion with Adaptive Feature Selection for Sparse Semantic Segmentation Network", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Feb. 8, 2021, pp. 12542–12551, DOI: 10.1109/CVPR46437.2021.01236 (cit. on pp. 24, IV).

[28] article Hung-Yueh Chiang, Yen-Liang Lin, Yueh-Cheng Liu, and Winston H. Hsu, "A Unified Point-Based Framework for 3D Segmentation", *in*: *International Conference on 3D Vision (3DV)*, Aug. 1, 2019, pp. 155–163, DOI: 10.1109/3DV.2019.00026 (cit. on pp. 24, IV).

[29] article J.M. Coughlan and A.L. Yuille, "Manhattan World: Compass Direction from a Single Image by Bayesian Inference", *in*: *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999, 941–947 vol.2, DOI: 10.1109/ICCV.1999.790349 (cit. on p. 110).

[30] article Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes", *in*: *30th IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, vol. 2017-Janua, Nov. 6, 2017, pp. 2432–2443, DOI: 10.1109/CVPR.2017.261 (cit. on pp. 21, 22, 26, 35, 91, IV).

[31] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi, *ProcTHOR: Large-Scale Embodied AI Using Procedural Generation*, June 14, 2022, DOI: 10.48550/arXiv.2206.06994, arXiv: 2206.06994 [cs], preprint (cit. on pp. 27, 141).

[32] article Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database", *in*: *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255, DOI: 10.1109/CVPR.2009.5206848 (cit. on p. 15).

[33] online Jing Du, Zuning Jiang, Shangfeng Huang, Zongyue Wang, Jinhe Su, Songjian Su, Yundong Wu, and Guorong Cai, "Point Cloud Semantic Segmentation Network Based on Multi-Scale Feature Fusion", *in*: *Sensors* **21**.*5* (5 Jan. 2021), p. 1625, DOI: 10.3390/s21051625 (cit. on pp. 24, III, IV).

[34] article Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe, "Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds", *in*: *IEEE International Conference on Computer Vision Workshops (ICCVW)*, Feb. 5, 2017, pp. 716–724, DOI: 10.1109/ICCVW.2017.90 (cit. on pp. 24, III).

[35] online Thomas A Feo and Mauricio G. C Resende, "A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem", *in*: *Operations Research Letters* **8**.*2* (Apr. 1, 1989), pp. 67–71, DOI: 10.1016/0167-6377(89)90002-3 (cit. on p. 100).

[36] online Duarte Fernandes, António Silva, Rafael Névoa, Cláudia Simões, Dibet Gonzalez, Miguel Guevara, Paulo Novais, João Monteiro, and Pedro Melo-Pinto, "Point-Cloud Based 3D Object Detection and Classification Methods for Self-Driving Applications: A Survey and Taxonomy", *in*: *Information Fusion* **68** (Apr. 1, 2021), pp. 161–191, DOI: 10.1016/j.inffus.2020.11.002 (cit. on pp. 18, 23, 140).

[37] article 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*, June 2012, pp. 3354–3361, DOI: 10.1109/CVPR.2012.6248074 (cit. on p. 22).

[38] online David Griffiths and Jan Boehm, "A Review on Deep Learning Techniques for 3D Sensed Data Classification", *in*: *Remote Sensing* **11**.*12* (June 25, 2019), p. 1499, DOI: 10.3390/rs11121499 (cit. on p. 16).

[39] online David Griffiths and Jan Boehm, "Weighted Point Cloud Augmentation for Neural Network Training Data Class-Imbalance", *in*: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **XLII-2/W13** (June 5, 2019), pp. 981–987, DOI: 10.5194/isprs-archives-XLII-2-W13-981-2019 (cit. on pp. 90, 146).

[40] online Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun, "Deep Learning for 3D Point Clouds: A Survey", *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**.*12* (2020), pp. 4338–4364, DOI: 10.1109/TPAMI.2020.3005434 (cit. on pp. 16–18, 23, 146).

[41] article Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D. Wegner, Konrad Schindler, and Marc Pollefeys, "Semantic3D.Net: A New Large-scale Point Cloud Classification Benchmark", *in*: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Apr. 12, 2017, pp. 91–98 (cit. on pp. 22, 26, 91, IV).

[42] article Timo Hackel, Jan D. Wegner, and Konrad Schindler, "Contour Detection in Unstructured 3D Point Clouds", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1610–1618 (cit. on pp. 44, 145).

[43] article Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla, "Understanding RealWorld Indoor Scenes with Synthetic Data", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4077–4085, DOI: 10.1109/CVPR.2016.442 (cit. on p. 67).

[44] article Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition", *in*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016-Decem, 2016, pp. 770–778, DOI: 10.1109/CVPR.2016.90 (cit. on p. 53).

[45] online Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition", *in*: *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**.*9* (June 18, 2014), pp. 346–361, DOI: 10.1007/978-3-319-10578-9_23 (cit. on p. 20).

[46] article Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham, "RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 11105–11114, DOI: 10.1109/CVPR42600.2020.01112 (cit. on pp. 24, 26, 145, III, IV).

[47] article Qiangui Huang, Weiyue Wang, and Ulrich Neumann, "Recurrent Slice Networks for 3D Segmentation of Point Clouds", *in*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1, 2018, pp. 2626–2635, DOI: 10.1109/CVPR.2018.00278 (cit. on pp. 19, 24, 147, III, IV).

[48] article Maximilian Jaritz, Jiayuan Gu, and Hao Su, "Multi-View PointNet for 3D Scene Understanding", *in*: *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019, pp. 3995–4003, DOI: 10.1109/ICCVW.2019.00494 (cit. on pp. 24, 26, IV).

[49] Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, and Cewu Lu, *PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation*, Nov. 24, 2018, arXiv: 1807.00652, preprint (cit. on pp. 19, 24, III, IV).

[50] online Hyungki Kim, Changmo Yeo, Inhwan Dennis Lee, and Duhwan Mun, "Deep-Learning-Based Retrieval of Piping Component Catalogs for Plant 3D CAD Model Reconstruction", *in*: *Computers in Industry* **123** (Dec. 1, 2020), p. 103320, DOI: 10.1016/j.compind.2020.103320 (cit. on p. 12).

[51] article Andrew King, Suchendra M Bhandarkar, and Brian M Hopkinson, "Deep Learning for Semantic Segmentation of Coral Reef Images Using Multi-View Information", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019, pp. 1–10 (cit. on p. 18).

[52] article Artem Komarichev, Zichun Zhong, and Jing Hua, "A-CNN: Annularly Convolutional Neural Networks on Point Clouds", *in*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. June-2019, June 1, 2019, pp. 7421–7430, DOI: 10.1109/CVPR.2019.00760 (cit. on pp. 19, 24, 26, III, IV).

[53] article Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *in*: *Neural Information Processing Systems Conference (NIPS)*, vol. 25, 2012, pp. 1–9 (cit. on p. 15).

[54] Ministère du travail du plein emploi et de l'insertion, *Industrie Pétrochimique*, Ministère du Travail, du Plein emploi et de l'Insertion, Mar. 21, 2011, URL: https://travail-emploi.gouv.fr/archives/archives-courantes/metiers-et-activites/article/industrie-petrochimique (visited on 02/24/2023) (cit. on p. 2).

[55] article Loic Landrieu and Martin Simonovsky, "Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4558–4567, DOI: 10.1109/CVPR.2018.00479 (cit. on pp. 24, 147, III).

[56] article Yann Lecun, Léon Bottou, Yoshua Bengio, and Parick Haffner, "Gradient-Based Learning Applied to Document Recognition", *in*: *Proceedings of the IEEE*, vol. 86, 1998, pp. 2278–2324, DOI: 10.1109/5.726791 (cit. on p. 15).

[57] article Mengtian Li, Yuan Xie, Yunhang Shen, Bo Ke, Ruizhi Qiao, Bo Ren, Shaohui Lin, and Lizhuang Ma, "HybridCR: Weakly-Supervised 3D Point Cloud Semantic Segmentation via Hybrid Contrastive Regularization", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14930–14939 (cit. on pp. 23, 24).

[58] article Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen, "PointCNN: Convolution On X-Transformed Points", *in*: *Advances in Neural Information Processing Systems*, vol. 31, 2018, pp. 820–830 (cit. on pp. 24, 26, III, IV).

[59] article Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Sai Bi, Hong-Xing Yu, Zexiang Xu, Kalyan Sunkavalli, Milos Hasan, Ravi Ramamoorthi, and Manmohan Chandraker, "OpenRooms: An Open Framework for Photorealistic Indoor Scene Datasets", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7190–7199 (cit. on p. 27).

[60] article Min Lin, Qiang Chen, and Shuicheng Yan, "Network in Network", *in*: *International Conference on Learning Representations (ICLR)*, 2014 (cit. on p. 20).

[61] article Zhi-Hao Lin, Sheng-Yu Huang, and Yu-Chiang Frank Wang, "Convolution in the Cloud: Learning Deformable Kernels in 3D Graph Convolution Networks for Point Cloud Analysis", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 1797–1806, DOI: 10.1109/CVPR42600.2020.00187 (cit. on pp. 26, 45, 110, 112, 114).

[62] article Jitao Liu, Songmin Dai, and Xiaoqiang Li, "PCCN:POINT Cloud Colorization Network", *in*: *IEEE International Conference on Image Processing (ICIP)*, Sept. 2019, pp. 3716–3720, DOI: 10.1109/ICIP.2019.8803633 (cit. on p. 28).

[63] article E. S. Malinverni, R. Pierdicca, M. Paolanti, M. Martini, C. Morbidoni, F. Matrone, and A. Lingua, "Deep Learning for Semantic Segmentation of 3D Point Cloud", *in*: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 42, 2/W15, Aug. 19, 2019, pp. 735–742, DOI: 10.5194/isprs-archives-XLII-2-W15-735-2019 (cit. on p. 21).

[64] article Daniel Maturana and Sebastian Scherer, "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition", *in*: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928, DOI: 10.1109/IROS.2015.7353481 (cit. on p. 17).

[65] article John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J. Davison, "SceneNet RGB-D: Can 5M Synthetic Images Beat Generic ImageNet Pre-training on Indoor Segmentation?", *in*: *IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2697–2706, DOI: 10.1109/ICCV.2017.292 (cit. on pp. 27, 49, 65).

[66] article Hsien-Yu Meng, Lin Gao, YuKun Lai, and Dinesh Manocha, "VV-Net: Voxel VAE Net with Group Convolutions for Point Cloud Segmentation", *in*: *IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2019, pp. 8499–8507, DOI: 10.1109/ICCV.2019.00859 (cit. on pp. 18, 24, 147).

[67] article Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation", *in*: *IEEE International Conference on Intelligent Robots and Systems (ICIRS)*, 2019, pp. 4213–4220, DOI: 10.1109/IROS40897.2019.8967762 (cit. on pp. 18, 23).

[68] online A. Nivaggioli, J. F. Hullo, and G. Thibault, "Using 3D Models to Generate Labels for Panoptic Segmentation of Industrial Scenes", *in*: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* **IV-2/W5** (May 29, 2019), pp. 61–68, DOI: 10.5194/isprs-annals-IV-2-W5-61-2019 (cit. on p. 141).

[69] article Florian Noichl, Alex Braun, and Andre Borrmann, ""BIM-to-Scan" for Scan-to-BIM: Generating Realistic Synthetic Ground Truth Point Clouds Based on Industrial 3D Models", *in*: *European Conference on Computing in Construction (EC3)*, July 26, 2021, pp. 164–172, DOI: 10.35490/EC3.2021.166 (cit. on pp. 27, 50).

[70] article Panagiotis Papadakis, "A Use-Case Study on Multi-View Hypothesis Fusion for 3D Object Classification", *in*: *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 2446–2452, DOI: 10.1109/iccvw.2017.288 (cit. on p. 35).

[71] Josh Patterson and Adam Gibson, *Deep Learning A Practitioner's Approach*, O'Reilly, Aug. 2017 (cit. on pp. 15, 119).

[72] article Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun, "Large Kernel Matters - Improve Semantic Segmentation by Global Convolutional Network", *in*: *30th IEEE Conference on Computer Vision and Pattern Recognition*

(CVPR), vol. 2017-Janua, Nov. 6, 2017, pp. 1743–1751, DOI: 10.1109/CVPR.2017.189 (cit. on p. 20).

[73] article Quang-Hieu Hieu Pham, Duc Thanh Nguyen, Binh-Son Hua Gemma Roig, Sai-Kit Kit Yeung, Thanh Nguyen, Binh Son Hua, Gemma Roig, and Sai-Kit Kit Yeung, "JSIS3D: Joint Semantic-Instance Segmentation of 3D Point Clouds with Multi-Task Pointwise Networks and Multi-Value Conditional Random Fields", *in*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019-June, June 1, 2019, pp. 8827–8836, DOI: 10.1109/CVPR.2019.00903 (cit. on pp. 24, III).

[74] article Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas, "Volumetric and Multi-View CNNs for Object Classification on 3D Data", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5648–5656 (cit. on pp. 17, 18).

[75] article Charles Ruizhongtai Qi, Hao Su, Mo Kaichun, and Leonidas J Guibas, "PointNet : Deep Learning on Point Sets for 3D Classification and Segmentation", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660 (cit. on pp. 16, 19, 23, 24, 89, 114, 140, III).

[76] article Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space", *in*: *31st International Conference on Neural Information Processing Systems (NIPS)*, June 7, 2017, pp. 5105–5114 (cit. on pp. 12, 19, 23, 24, 53, 140, 146, IV).

[77] article Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger, "OctNet: Learning Deep 3D Representations at High Resolutions", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Nov. 15, 2017, pp. 3577–3586 (cit. on p. 17).

[78] article Hayko Riemenschneider, A. Bódis-Szomorú, Julien Weissenberg, and L. Gool, "Learning Where to Classify in Multi-view Semantic Segmentation", *in*: *European Conference on Computer Vision (ECCV)*, 2014, DOI: 10.1007/978-3-319-10602-1_34 (cit. on p. 21).

[79] article Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", *in*: *Lecture Notes in Computer Science*, vol. 9351, May 18, 2015, pp. 234–241, DOI: 10.1007/978-3-319-24574-4_28 (cit. on p. 20).

[80] article Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa, "Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation", *in*: *IEEE/CVF Conference on Computer Vision and Pat-*

tern Recognition (CVPR), June 2018, pp. 3752–3761, DOI: 10.1109/CVPR.2018.00395 (cit. on p. 27).

[81] article Evan Shelhamer, Jonathan Long, and Trevor Darrell, "Fully Convolutional Networks for Semantic Segmentation", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440, DOI: 10.1109/CVPR.2015.7298965 (cit. on p. 20).

[82] article Takayuki Shinohara, Haoyi Xiu, and Masashi Matsuoka, "Point2color: 3D Point Cloud Colorization Using a Conditional Generative Network and Differentiable Rendering for Airborne LiDAR", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2021, pp. 1062–1071, DOI: 10.1109/CVPRW53098.2021.00117 (cit. on p. 28).

[83] article Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus, "Indoor Segmentation and Support Inference from RGBD Images", *in*: *European Conference on Computer Vision (ECCV)*, vol. 7576, 2012, pp. 746–760, DOI: 10.1007/978-3-642-33715-4_54 (cit. on pp. 21, 22).

[84] article Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *in*: *3rd International Conference on Learning Representations (ICLR)*, 2015, pp. 1–14 (cit. on p. 20).

[85] article Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao, "SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 567–576, DOI: 10.1109/CVPR.2015.7298655 (cit. on pp. 21, 22).

[86] online Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *in*: *Journal of Machine Learning Research* **15** (2014), pp. 1929–1958 (cit. on p. 145).

[87] article Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller, "Multi-View Convolutional Neural Networks for 3D Shape Recognition", *in*: *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 945–953, DOI: 10.1109/ICCV.2015.114 (cit. on p. 18).

[88] article Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning", *in*: *31st AAAI Conference on Artificial Intelligence (AAAI)*, Feb. 23, 2017, pp. 4278–4284 (cit. on p. 20).

[89] article Alessandro Tasora, Radu Serban, Hammad Mazhar, Arman Pazouki, Daniel Melanz, Jonathan Fleischmann, Michael Taylor, Hiroyuki Sugiyama, and Dan Negrut, "Chrono: An Open Source Multi-physics Dynamics Engine", *in*: *High Performance Computing in Science and Engineering*, 2016, pp. 19–49, DOI: 10.1007/978-3-319-40361-8_2 (cit. on p. 49).

[90] article Lyne Tchapmi, Christopher Choy, Iro Armeni, Junyoung Gwak, and Silvio Savarese, "SEGCloud: Semantic Segmentation of 3D Point Clouds", *in*: *International Conference on 3D Vision (3DV)*, May 25, 2017, pp. 537–547, DOI: 10.1109/3DV.2017.00067 (cit. on pp. 18, 24).

[91] Hugues Thomas, "Learning New Representations for 3D Point Cloud Semantic Segmentation", PhD thesis, Université Paris sciences et lettres, Nov. 19, 2019 (cit. on pp. 94, 145).

[92] article Hugues Thomas, Charles R. Qi, Jean Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas Guibas, "KPConv: Flexible and Deformable Convolution for Point Clouds", *in*: *IEEE International Conference on Computer Vision (ICCV)*, vol. 2019-Octob, Oct. 1, 2019, pp. 6410–6419, DOI: 10.1109/ICCV.2019.00651 (cit. on pp. 19, 24, 26, 45, 53, 93, 113, 114, 146, III).

[93] article Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan, "Graph Attention Convolution for Point Cloud Semantic Segmentation", *in*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10288–10297, DOI: 10.1109/CVPR.2019.01054 (cit. on pp. 19, 24, III, IV).

[94] online Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong, "O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis", *in*: *ACM Transactions on Graphics (TOG* **36**.*4* (2017), p. 72, DOI: 10.1145/3072959.3073608 (cit. on p. 17).

[95] article Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann, "SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation", *in*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Dec. 14, 2018, pp. 2569–2578, DOI: 10.1109/CVPR.2018.00272 (cit. on pp. 24, III).

[96] article Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia, "Associatively Segmenting Instances and Semantics in Point Clouds", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4091–4100 (cit. on pp. 24, III).

[97] article Xu Wang, Yuyan Li, and Ye Duan, "Fast Point Voxel Convolution Neural Network with Selective Feature Fusion for Point Cloud Semantic Segmentation", *in*: *Lecture Notes in Computer Science*, Sept. 23, 2021, DOI: 10.1007/978-3-030-90439-5_25 (cit. on pp. 24, III).

[98] article Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon, "Dynamic Graph Cnn for Learning on Point Clouds", *in*: *ACM Transactions on Graphics*, vol. 38, 5, Oct. 1, 2019, DOI: 10.1145/3326362 (cit. on pp. 19, 24, III).

[99] online Lukas Winiwarter, Alberto Manuel Esmorís Pena, Hannah Weiser, Katharina Anders, Jorge Martínez Sánchez, Mark Searle, and Bernhard Höfle, "Virtual Laser Scanning with HELIOS++: A Novel Take on Ray Tracing-Based Simulation of Topographic Full-Waveform 3D Laser Scanning", *in*: *Remote Sensing of Environment* **269** (Feb. 1, 2022), p. 112772, DOI: 10.1016/j.rse.2021.112772 (cit. on pp. 50, 144).

[100] article Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer, "SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud", *in*: *IEEE International Conference on Robotics and Automation (ICRA)*, Sept. 10, 2018, pp. 1887–1893, DOI: 10.1109/ICRA.2018.8462926 (cit. on pp. 18, 23, 27).

[101] article Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer, "SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud", *in*: *International Conference on Robotics and Automation (ICRA)*, May 1, 2019, pp. 4376–4382, DOI: 10.1109/ICRA.2019.8793495 (cit. on pp. 27, 145).

[102] article Wenxuan Wu, Zhongang Qi, and Li Fuxin, "PointCONV: Deep Convolutional Networks on 3D Point Clouds", *in*: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019-June, June 1, 2019, pp. 9621–9630, DOI: 10.1109/CVPR.2019.00985 (cit. on pp. 24, IV).

[103] article Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu, "Transfer Learning from Synthetic to Real LiDAR Point Cloud for Semantic Segmentation", *in*: *AAAI Conference on Artificial Intelligence (AAAI)*, vol. 36, 3, June 28, 2022, pp. 2795–2803, DOI: 10.1609/aaai.v36i3.20183 (cit. on pp. 27, 29, 84, 139, 144).

[104] online Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu, "Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation", *in*: *IEEE Geoscience and remote sensing magazine* **8** (2019), pp. 38–59, DOI: 10.1109/MGRS.2019.2937630 (cit. on pp. 14, 16).

[105] Yunhao Yang, Yuhan Zheng, Yi Wang, and Chandrajit L. Bajaj, *Deep Contrastive Patch-Based Subspace Learning for Camera Image Signal Processing*, Apr. 1, 2021, preprint (cit. on p. 144).

[106] article Xiaoqing Ye, Jiamao Li, Hexiao Huang, Liang Du, and Xiaolin Zhang, "3D Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation", *in*: *European Conference on Computer Vision (ECCV)*, 2018, pp. 403–417 (cit. on pp. 19, 24, 147, III, IV).

[107] online Changmo Yeo, Seyoon Kim, Hyungki Kim, Siro Kim, and Duhwan Mun, "Deep Learning Applications in an Industrial Process Plant: Repository of Segmented Point Clouds for Pipework Components", *in*: *JMST Advances* **2**.*1* (Mar. 1, 2020), pp. 15–24, DOI: 10.1007/s42791-019-00027-y (cit. on pp. 12, 63).

[108] online Chao Yin, Boyu Wang, Vincent J. L. Gan, Mingzhu Wang, and Jack C. P. Cheng, "Automated Semantic Segmentation of Industrial Point Clouds Using ResPointNet++", *in*: *Automation in Construction* **130** (Oct. 1, 2021), p. 103874, DOI: 10.1016/j.autcon.2021.103874 (cit. on pp. 12, 142, 143).

[109] article Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia, "PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing", *in*: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5560–5568 (cit. on pp. 19, 24, III).

[110] article Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, "Pyramid Scene Parsing Network", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, DOI: 10.1109/CVPR.2017.660 (cit. on p. 20).

[111] article Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3D ShapeNets: A Deep Representation for Volumetric Shapes", *in*: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1912–1920, DOI: 10.1109/CVPR.2015.7298801 (cit. on p. 112).

COLLEGE MATHSTIC
DOCTORAL BRETAGNE
BRETAGNE OCEANE

ÉCOLE NATIONALE D'INGÉNIEURS DE BREST

Titre : Amélioration de l'informativité des données pour la segmentation sémantique, par apprentissage profond, de nuage de points représentant des scènes industrielles.

Mot clés : Apprentissage profond, segmentation sémantique, nuage de points, industrie 4.0

Résumé : Grâce aux techniques actuelles, la reconstruction de notre environnement sous forme d'espace 3D est facilité. Cependant, transformer une installation industrielle existante en modèle d'information de la construction (BIM) via un logiciel de Conception Assisté par Ordinateur (CAO) reste une tâche fastidieuse. Le but de cette thèse est d'améliorer les performances de segmentation sémantique de nuage de points appliquée à un environnement industriel. L'état de l'art a montré l'existence de méthodes achevant de bonnes performances de segmentation sur les bases de données disponible publiquement mais aussi l'absence d'une telle base pour le domaine industriel. Une première étude des données nous étant disponible a permis de montrer les particularités et difficultés propre au travail sur données d'origine industrielle. Ces particularités, combinées à un manque de données, à diriger nos recherches vers des méthodes permettant d'utiliser au mieux les données disponibles. Pour résoudre cette tâche, deux pistes furent explorées. Premièrement en créant une méthode de génération de données synthétiques qui mena à une étude entre réalisme des données synthétiques et performances de segmentation. Deuxièmement par l'étude de différentes méthodes de transformation de données applicables avant la couche d'entrée du réseau de segmentation. La capacité de certaines de ces méthodes à améliorer les résultats de segmentation est démontrée.

Title: Enhancing data informativeness in deep-learning based, point cloud semantic segmentation of industrial scenes.

Keywords: Deep learning, semantic segmentation, point cloud, industry 4.0

Abstract: Thanks to current technologies, reconstructing our environments as a 3D spaces is easier. However, transforming existing industrial facilities to Building Information Model (BIM) via Computer Assisted Drawing (CAD) software is still a tedious task. The goal of this thesis is to improve performance of point cloud semantic segmentation applied to an industrial setting. The literature review showed that current state of the art methods achieve high segmentation results on publicly available dataset but no such dataset exist for an industrial setting. A first study of available data allowed us to show the specificity and difficulties in working with industrial data. These peculiarities, combined with a lack in data quantity, heads this thesis towards finding ways to use available data more effectively. To solve this task, two ideas were explored. First, a synthetic data generation method was created and the relationship between synthetic data realism and segmentation performance is studied. Secondly, different data transformation to be applied before the segmentation network input are studied and the ability of some of them to increase segmentation performance is shown.