Semantic segmentation for 3D change detection in urban and railway context using LiDAR point clouds

Dissertation presented in partial fulfillment of the requirements for the degree of

Doctor of Science (PhD) by

Abderrazzaq Kharroubi

University of Liège

Faculty of Science, Doctoral College in Geography

July 2025

This dissertation is approved by:

Prof. Dr. Roland Billen	Supervisor
Prof. Dr. Rafika Hajji	Co-supervisor

Composition of the doctoral committee:

Prof. Dr. Roland Billen	University of Liège, Belgium
Prof. Dr. Rafika Hajji	IAV Hassan II, Morocco
Prof. Dr. Fabio Remondino	3DOM Trento, Italy
Prof. Dr. Gilles Louppes	University of Liège, Belgium
Dr. Florent Poux	3D Geodata Academy, France

Composition of the examination committee:

Prof. Dr. Gilles Louppes	University of Liège, Belgium (President)
Prof. Dr. Roland Billen	University of Liège, Belgium (Supervisor)
Prof. Dr. Rafika Hajji	IAV Hassan II, Morocco (Co-supervisor)
Dr. Florent Poux	3D Geodata Academy, France
Prof. Dr. Roberta Ravanelli	University of Liège, Belgium (Secretary)
Prof. Dr. Fabio Remondino	3DOM Trento, Italy
Prof. Dr. Alain De Wulf	Ghent University, Belgium

This research was conducted at the GeoScITY Lab of the Geomatics Unit, part of the University of Liège, from October 2021 to March 2025. It was financed by the National Fund for Scientific Research (FNRS).

ABSTRACT

The analysis of 3D changes in complex environments, such as urban and railway infrastructures, is essential for applications including digital twins, infrastructure monitoring, and land management. While multi-temporal point clouds provide rich spatial and temporal information, their unstructured nature, lack of semantics, and acquisition inconsistencies make automated change detection a challenging task. This thesis addresses these challenges by proposing a comprehensive object-based framework that integrates semantic segmentation, change indicators, and structured modeling to improve the reliability, interpretability, and interoperability of 3D change detection.

The thesis reviews existing 3D change detection methods, highlighting key challenges related to noise sensitivity, lack of semantic interpretation, and limited integration into structured outputs. In response, a semantic and geometric fusion method is introduced. Bi-temporal point clouds are first semantically segmented, followed by Cut-Pursuit clustering to extract spatially coherent object instances. Second, object-level change indicators, including geometric descriptors such as verticality, sphericity, and omnivariance, are computed and used to classify changes into four categories: appeared, disappeared, modified, and unchanged. Third, a Random Forest classifier is trained to predict change labels, demonstrating improved spatial coherence and robustness to noise compared to point-wise approaches.

To support practical applications in city modeling and infrastructure documentation, the thesis introduces a rule-based workflow that formalizes detected changes into a CityJSON representation. This structured output encodes temporal, geometric, and semantic attributes of changed objects without requiring labeled training data. The approach allows for efficient updates of digital twins and facilitates integration into 3D geographic information systems. The framework is also extended to the monitoring of railway environments. A novel annotated dataset, Rail3D, is created to support semantic segmentation of railway-specific classes. Deep learning models are benchmarked on this dataset, and the results are used to detect and analyze vegetation changes over time. This component highlights the applicability of the proposed methods for infrastructure maintenance and monitoring tasks.

This thesis demonstrates how semantic enrichment, geometric analysis, and structured representation can be effectively combined to advance the state of 3D change detection. The developed methods are validated on real-world applications and contribute to the automation and interpretability of 3D point cloud analysis in dynamic environments.

Keywords: 3D change detection, point cloud, LiDAR, segmentation, classification, object-based, clustering, Cut-Pursuit, CityJSON, digital twin, geometry, semantics, railway, vegetation, buildings, ground, bi-temporal, feature extraction, annotation, supervised learning, rule-based, dataset, modeling, automation.

RÉSUMÉ

L'analyse du changement 3D dans des environnements complexes, tels que les infrastructures urbaines et ferroviaires, est essentielle pour des applications telles que les jumeaux numériques, la surveillance des infrastructures et la gestion du territoire. Malgré la richesse d'informations spatiale et temporelle qu'offrent les nuages de points multi-temporels, leur nature non structurée, leur ambiguïté sémantique et les variations liées aux acquisitions rendent la détection automatisée des changements particulièrement difficile. Ce travail de doctorat propose un processus complet, basé sur les objets, qui intègre la segmentation sémantique, des descripteurs géométriques et une modélisation structurée afin d'améliorer la fiabilité, l'interprétabilité et l'interopérabilité de la détection de changement en 3D.

La thèse revoit les méthodes existantes de détection de changement 3D, en mettant en évidence les principales limites liées à la sensibilité au bruit, au manque d'interprétation sémantique, et à l'absence d'intégration dans des structures standardisées. En réponse, un cadre de fusion sémantico-géométrique est proposé. Les nuages de points bi-temporels sont d'abord segmentés sémantiquement, puis regroupés en objets cohérents à l'aide de l'algorithme de partitionnement Cut-Pursuit. Des indicateurs de changement au niveau des objets, incluant des descripteurs géométriques tels que la verticalité, la sphéricité et l'omnivariance, sont extraits puis utilisés pour classer les changements en quatre catégories : apparus, disparus, modifiés et inchangés. Un classifieur par forêts aléatoires est entraîné pour prédire ces étiquettes, avec des performances supérieures aux approches point-à-point en termes de cohérence spatiale et de robustesse au bruit.

Pour une application concrète dans le cadre de la modélisation urbaine et de la documentation des infrastructures, un processus basé sur des règles est introduit pour formaliser les changements détectés au format CityJSON. Cette représentation structurée encode les attributs temporels, géométriques et sémantiques des objets changés sans nécessiter de données annotées. Cette structuration permet une mise à jour efficace des jumeaux numériques et une intégration aisée dans les systèmes d'information géographique 3D. Le travail est ensuite étendu à la surveillance des environnements ferroviaires. Un nouveau jeu de données annoté, Rail3D, est créé pour permettre la segmentation sémantique de classes spécifiques aux voies ferrées. Des modèles d'apprentissage profond sont évalués sur ce jeu de données, et les résultats sont utilisés afin de détecter et analyser les changements liés à la végétation au fil du temps. Cette application illustre la pertinence de la méthode proposée pour des tâches de surveillance et de maintenance d'infrastructures.

Les contributions de cette thèse montrent comment l'enrichissement sémantique, l'analyse géométrique et la représentation structurée peuvent être combinés efficacement pour faire progresser l'état de l'art en détection de changement 3D. Les méthodes développées sont validées sur des jeux de données simulés et réels et contribuent à l'automatisation et à l'interprétation des analyses de nuages de points dans des environnements dynamiques.

Mots clés : Détection de changement 3D, nuage de points, LiDAR, segmentation, classification, objets, clustering, Cut-Pursuit, CityJSON, jumeau numérique, géométrie, sémantique, ferroviaire, végétation, bâtiments, sol, multi-temporel, extraction de caractéristiques, annotation, apprentissage supervisé, règles, jeu de données, modélisation, automatisation.

ACKNOWLEDGEMENT

I wish to express my gratitude to those who supported me during my PhD.

First and foremost, I thank my thesis supervisor, Prof. Roland Billen, and co- supervisor, Prof. Rafika Hajji, for their constant guidance, expertise, and trust in my work. Their feedback and encouragement were essential to the success of this research. I am also grateful to my thesis committee members: Prof. Fabio Remondino, Prof. Gilles Louppes, and Dr. Florent Poux, for their valuable feedback and constructive advice in this process.

I would also like to warmly thank the members of my thesis jury, Prof. Dr. Roberta Ravanelli (University of Liège) and Prof. Dr. Alain De Wulf (Ghent University), for their time, interest in my work, and participation in the defense.

I extend my thanks to the University of Liège for providing the resources and academic environment that made this work possible. I also gratefully acknowledge the financial support received from the Fonds National de la Recherche Scientifique (FNRS), which enabled me to fully dedicate myself to this PhD research.

To the mentors, colleagues, and peers who contributed directly to my work, I acknowledge your impact and am thankful for the knowledge and inspiration I gained from our interactions.

I am fortunate to have shared these years with supportive friends and colleagues. Special thanks to Zouhair, Imane, Anass, Charline, Thomas, and all others for their friendship, shared struggles, and moments of joy.

To my wife, I owe endless gratitude for your patience, understanding, and support during the challenges of the years. You have been my anchor this entire time. I thank my mother, brothers, sister, and entire family for their lifelong encouragement and belief in my goals. Without your love and sacrifices, this achievement would not have been possible. I also dedicate a special thought to the memory of my father, whose values and efforts continue to guide me.

While I cannot name everyone individually, I recognize the many individuals (administrative staff, collaborators, students, and fellow researchers) who contributed to this work in small but meaningful ways. This thesis is the result of many contributions, and I am humbled by the collective effort behind it.

Abderrazzaq Kharroubi, 2025.

CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Full form
2D	Two-Dimensional
3D	Three Dimensional
3DCD	Three Dimensional Change Detection
AHN	Actueel Hoogtebestand Nederland (Dutch National Height Dataset)
AI	Artificial Intelligence
ALS	Airborne Laser Scanning
CNN	Convolutional Neural Network
CRF	Conditional Random Field
C2C	Cloud-to-Cloud
C2C-2D	Cloud-to-Cloud 2D
DEM	Digital Elevation Model
DL	Deep Learning
DoD	DEM of Difference
DSM	Digital Surface Model
DTM	Digital Terrain Model
EPSG	European Petroleum Survey Group (Coordinate System)
FCNN	Fully Convolutional Neural Network
FNRS	Fonds National de la Recherche Scientifique (Belgian research funder)
FPFH	Fast Point Feature Histograms
GCN	Graph Convolutional Network
GeoScITY	Geospatial Data Science and City Information Modelling
GIS	Geographic Information Systems

INFRABEL	Belgian railway infrastructure manager
IoU	Intersection over Union
KPConv	Kernel Point Convolution
LAS	LASer file format (LiDAR data standard)
LAZ	Compressed LAS file format
LiDAR	Light Detection and Ranging
LoD	Level of Detail
LODetection	Level of Detection (uncertainty metric)
M3C2	Multi-Scale Model-to-Model Cloud Comparison
mIoU	mean Intersection over Union
MLS	Mobile Laser Scanning
MMS	Mobile Mapping System
ML	Machine Learning
nDSM	normalized Digital Surface Model
NFD	Nearest Feature Difference
NNS	Nearest Neighbour Search
OBCD	Object-Based Change Detection
ODbL	Open Database License
PCA	Principal Component Analysis
PICC	Plan d'Information Cadastral et Cartographique
PLS	Permanent Laser Scanner
Poisson	Poisson Surface Reconstruction
RANSAC	Random Sample Consensus
Rail3D	Railway 3D dataset

RF	Random Forest
ROI	Region of Interest
RQ	Research Question
SAM	Segment Anything Model
SCD	Semantic Change Detection
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle
ULS	Unmanned Laser Scanning
Z+F	Zoller + Fröhlich (LiDAR sensor manufacturer)

"No man ever steps in the same river twice,
for it is not the same river and he is not the same man"
Heraclitus

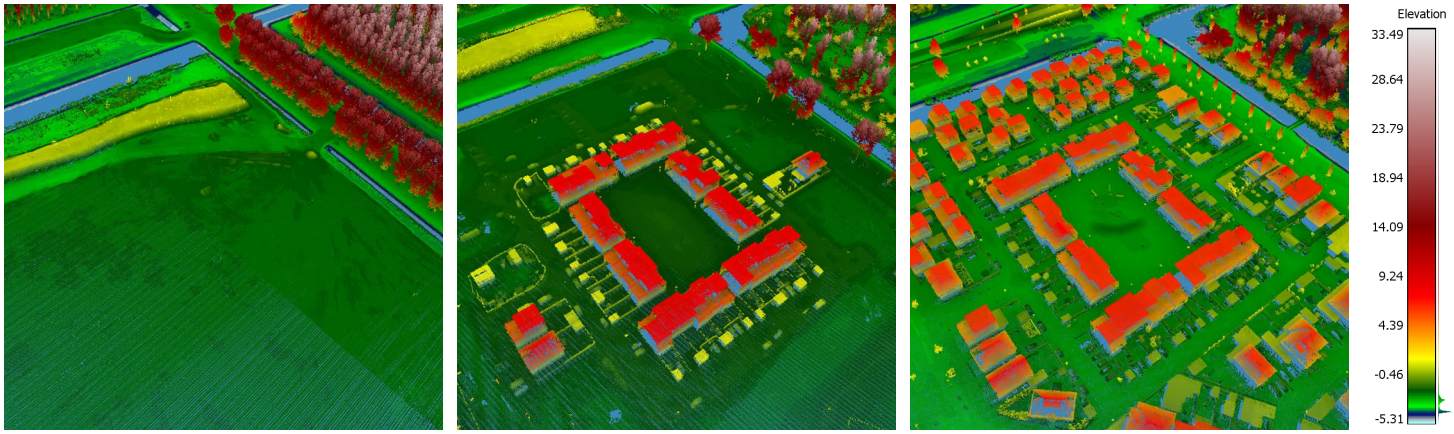
INTRODUCTION

Understanding how environments evolve over time is a critical need across various domains, from urban planning and infrastructure maintenance to environmental monitoring and digital twin development. As 3D sensing technologies have matured and become more accessible, they have enabled routine acquisition of large-scale point cloud datasets representing real-world scenes in high detail. While static analysis of such data has seen significant progress, detecting and understanding changes over time remains a complex and active research area. This thesis explores methods and applications of 3D change detection using bi-temporal point clouds, with a particular emphasis on the integration of semantic and geometric information. The following sections present the context and motivations, research questions, and the structure of the dissertation.

Context

Over the past decade, the widespread availability of 3D sensors, ranging from airborne LiDAR and terrestrial laser scanners to mobile mapping systems, has enabled the collection of detailed and dense 3D representations of our world. These 3D point clouds provide a precise geometric description of the scene at the moment of acquisition, offering a rich geospatial foundation for applications in urban planning, infrastructure monitoring, environmental analysis, and digital twin development [1], [2].

However, in dynamic environments where the built or natural scene changes over time, the ability to compare successive point clouds and detect what has changed becomes a critical challenge. Change detection (CD) using multi-temporal data aims to identify and characterize objects that have appeared, disappeared, or been modified between two acquisition epochs or more (**Figure 1**). Despite its potential, this task remains complex. Point clouds are unstructured, unordered, and non-uniform in density. Changes in acquisition parameters, such as angle, occlusion, noise, and temporal gaps, introduce variability that complicates direct comparison and increase the risk of false detections.



AHN2 AHN3 AHN5

Figure 1. Three point cloud samples from three different acquisition epochs ©AHN

Traditional 2D change detection techniques based on imagery are well-established and widely used in remote sensing [3], [4], [5]. In contrast, the transition to 3D brings unique challenges, particularly in terms of geometric complexity and data irregularity. These challenges are further compounded by the lack of standardized object definitions and structured outputs, which limits the use of 3D change results in downstream applications such as city model updates, infrastructure analysis, or automated asset monitoring.

In this context, object-based change detection has emerged as a promising approach. Rather than focusing on individual points or fixed voxel grids, object-based strategies aim to segment and analyze meaningful structures (such as buildings, trees, or ground surfaces) and to track their evolution across time (**Figure 2**). When these objects are enriched with semantic information, the resulting 3D change maps become more informative and interpretable, facilitating integration with geographic information systems (GIS) and semantic city modeling frameworks.

A key aspect of this thesis is to build upon the significant progress achieved in 3D semantic segmentation in recent years [6], where deep learning models allow accurate classification, even in complex and cluttered environments. These advancements can serve as a valuable foundation for object-based change detection. By incorporating semantic segmentation into the detection process, it becomes possible to improve object matching, reduce ambiguity in change labeling, and move towards more structured and interpretable outputs.

Nonetheless, the integration of semantic and geometric analysis for change detection remains a non-trivial task. First, semantic segmentation of point clouds itself is demanding, requiring either large, annotated datasets for supervised learning or robust unsupervised clustering algorithms. Second, reliable object-level matching across time requires similarity measures that are invariant to partial occlusions, viewpoint changes, or missing data. Third, assigning meaningful change categories, such as "building extension" or "vegetation loss", requires combining geometric evidence with semantic context and often requires some degree of rule-based reasoning.

Furthermore, there is a growing need to move beyond binary change maps towards multi-level representations that offer a more complete understanding of the scene. This includes not only detecting what changed, but also characterizing the nature, extent, and relevance of the change. In urban applications, such structured change information is crucial for updating 3D models, managing assets, and supporting decision-making processes in digital twins. Encoding changes using structured, interoperable formats such as CityJSON also enables longer-term integration and reuse across systems.

At the research level, there is a clear gap in scalable and transferable pipelines for 3D change detection that integrate semantic segmentation, geometric analysis, and structured modeling [7], [8]. Many existing approaches prioritize performance metrics without offering interpretability or generalizability. Others rely heavily on handcrafted rules that are not robust across different contexts or data sources. There is therefore a need for hybrid methodologies that combine the strengths of data-driven learning and

structured reasoning, while ensuring scalability to large-scale point clouds and compatibility with open geospatial standards.

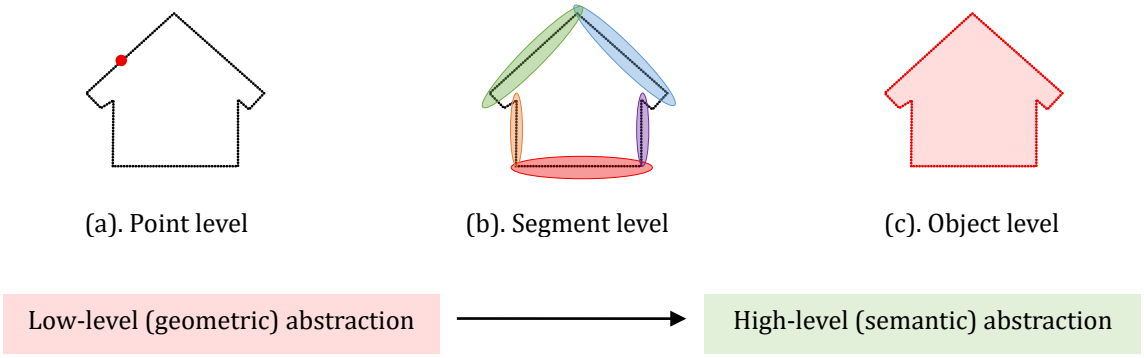


Figure 2. Levels of granularity in 3D change detection

This thesis responds to these challenges by proposing object-based methods for 3D change detection that integrate semantic information, geometric descriptors, and structured representation of changes. It explores both supervised and unsupervised strategies, evaluates the role of semantic segmentation in assisting detection, and introduces workflows for organizing and encoding change results into interoperable formats. The contributions are validated on datasets representing diverse environments: including synthetic urban scenes, aerial acquisitions, and railway networks, with an emphasis on practical utility, reproducibility, and relevance to real-world applications.

By formalizing change detection as a structured, object-based task that uses semantic information, this research contributes towards making 3D point clouds more actionable for dynamic scene understanding. It also opens new directions for multi-level change analysis, where changes are characterized at different granularities, from localized point-level variations to high-level semantic object abstraction, and offers new possibilities for monitoring, modeling, and managing dynamic 3D environments.

Research questions

At the intersection of semantic segmentation and geometric analysis, this dissertation explores the development of reliable, interpretable, and object-based change detection workflows using 3D point clouds. The main research question driving this thesis is:

How can we combine semantic segmentation and geometric analysis to detect, classify, and structure changes in 3D point clouds?

This overarching question is addressed through the following three complementary sub-questions:

- How can we reliably detect and characterize changes in 3D environments using bi-temporal point cloud data?
- How can semantic and geometric features be fused at the object level to improve change classification accuracy?
- How can detected changes be formalized into structured, interpretable, and interoperable models suitable for urban and railway applications?

These questions reflect the methodological, algorithmic, and application-oriented challenges tackled in the thesis, including the use of unsupervised clustering, deep learning architectures, and semantic structuring using standards such as CityJSON.

Thesis structure

This thesis is structured around four main chapters, which answer the research sub-questions and contributing to the development of 3D change detection using point clouds (**Figure 3**). Each chapter includes a preface to contextualize the contribution and is based on a journal or a conference publication adapted for integration within this document.

Chapter 1 reviews the state of the art in 3D change detection using point clouds. This chapter provides a detailed synthesis of existing methods for detecting changes in 3D scenes using multi-temporal point clouds. It begins with a suggested classification of standard and learning-based approaches, organized by methodological categories, application domain, and change units (pixel, point, voxel, segment, object). Then, it discusses core challenges such as registration, occlusion, and 3D data specificities. The objective is to critically examine current practices and motivate the design of a new object-based approach that better handles geometric and semantic information while offering structured outputs. The chapter serves as the theoretical foundation for the contributions developed in the following chapters.

Chapter 2 proposes a semantic-geometric object-based method for change detection. It introduces a complete workflow for detecting and labeling changes at the object level using classified and segmented point clouds. It leverages geometric descriptors, nearest neighbor search, and a machine learning classifier to classify objects as appeared, disappeared, unchanged, and modified objects. The supervised approach was evaluated on a simulated urban dataset designed to mimic real acquisition noise and occlusions. Evaluation metrics are proposed to assess change type classification, and a comparison is made with a point-based method baseline. This chapter contributes to the research by exploring how combining semantic and geometric features can improve the reliability of 3D change detection.

Chapter 3 develops a rule-based method for structured change detection and modeling. In scenarios where change training data is unavailable, this chapter introduces an

approach for detecting and structuring changes at the semantic object level. It relies on instance segmentation followed by geometric comparison and logical rules to categorize changes. In addition, the chapter proposes a methodology to transform raw detection results into a structured 3D city model using the CityJSON format. A new CityJSON extension is developed to encode change information in a standardized and interoperable form. This approach facilitates the integration of change information into urban digital twins and other smart city applications.

Chapter 4 investigates semantic segmentation of railway scenes for vegetation change detection. Focusing on railway infrastructure, this chapter uses supervised deep learning approaches for semantic segmentation of mobile mapping point clouds. A new annotated dataset is introduced and used to benchmark different neural architectures, including the Kernel Point Convolution (KPConv). The segmentation results provide class-level information such as vegetation, tracks, poles, and built structures. A subsequent section explores how these semantic labels are reused for detecting changes in vegetation over time. The chapter illustrates how high-quality semantic segmentation is integrated into change detection pipelines and adapted to domain-specific constraints.

Finally, the conclusion chapter summarizes the answers provided to the research questions, highlights the contributions made in each chapter, and discusses current limitations. It reflects on the implications of object-based change detection, the challenges of deploying such systems in operational contexts, and the potential for generalization to other types of 3D data and domains. It concludes by outlining future directions, including multi-level change detection, learning based object matching and tracking, and scene completion for occlusion handling. **Figure 3** presents the research sub-questions and their corresponding chapters.

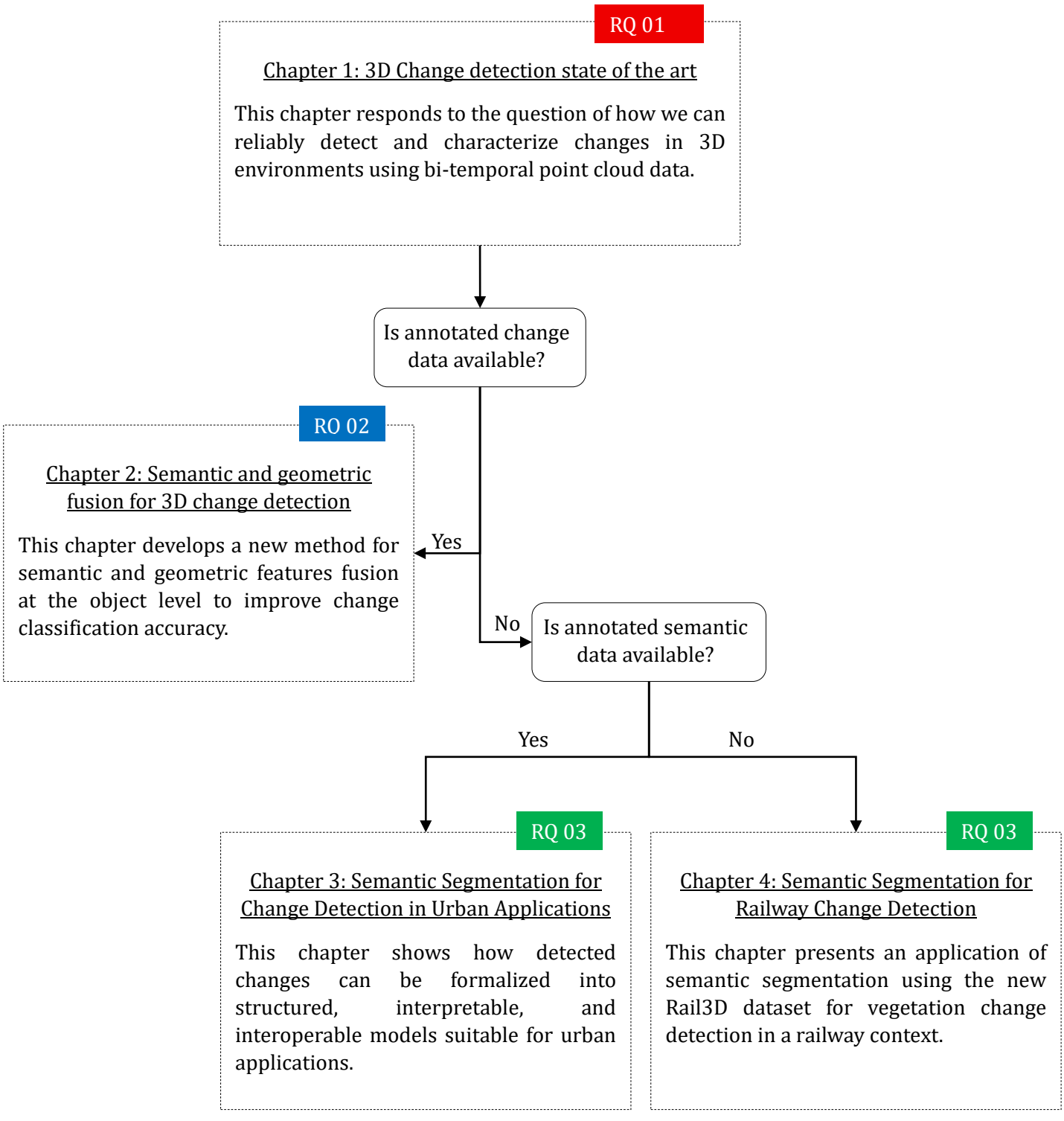


Figure 3. Overview of the thesis structure and links between the research sub-questions (RQ) and chapters.

CHAPTER 01:

3D Change Detection State of the Art

Preface

This chapter stems from work initiated in 2022, during the early stages of the thesis, to address the question: ***How can we reliably detect and characterize changes in 3D environments using multi-temporal point cloud data?***

At that time, the increasing availability of dense LiDAR acquisitions from airborne, terrestrial, and mobile platforms had already paved the way for richer 3D representations of the built and natural environment. Yet, identifying changes in these data, especially under conditions of occlusion, varying density, and limited ground truth, remained a major challenge. The need for robust, scalable, and interpretable 3D change detection (CD) techniques was particularly evident in domains such as urban reconstruction, vegetation monitoring, and railway infrastructure analysis.

The idea behind this chapter was to consolidate and critically review the literature on 3D CD, with a special emphasis on methods using point clouds. Initially, we observed that most existing reviews were either focused on 2D remote sensing (image-based approaches) or limited to a single application field. There was a clear lack of an integrated synthesis of standard, machine learning, and deep learning methods.

Thus, the first step was to explore the historical evolution of 3D CD algorithms and understand the main taxonomies proposed in the literature: pixel-based, point-based, voxel-based, segment-based, and object-based approaches. We also surveyed recent works that integrate semantic segmentation with change detection, as well as clustering techniques for object-level analysis. Our goal was not only to classify the methods but to highlight their assumptions, capabilities, and limitations, particularly with respect to registration errors, occlusions, point cloud heterogeneity, and the nature of changes being detected (e.g., appearance, disappearance, modification).

The chapter is based on the article "Three Dimensional Change Detection Using Point Clouds: A Review" (Kharroubi et al., 2022), which aimed to offer a systematic overview of state-of-the-art 3D CD methods, from traditional techniques (e.g., C2C, M3C2) to recent deep learning ones (e.g., Siamese KPConv, SiamPointNet++). However, this chapter goes beyond the original publication by adding an update of the latest advances in the field between 2022 and the writing of the thesis. This includes newer benchmark datasets, and emerging trends such as weakly supervised learning and 4D LiDAR analysis, ensuring the review remains both relevant and reflective of the current state of the art. Beyond the academic scope of the article, this chapter has been extended to connect with the subsequent chapters of the thesis. It lays the groundwork for the following work on object-based change detection.

Based on the article (Kharroubi et al. 2022)

Three Dimensional Change Detection Using Point Clouds: A Review

Kharroubi A, Poux F, Ballouch Z, Hajji R, Billen R. Geomatics. 2022; 2(4):457-485.

Abstract: Change detection is an important step for the characterization of object dynamics at the earth's surface. In multi-temporal point clouds, the main challenge is to detect true changes at different granularities in a scene subject to significant noise and occlusion. To better understand new research perspectives in this field, a deep review of recent advances in 3D change detection methods is needed. To this end, we present a comprehensive review of the state of the art of 3D change detection approaches, mainly those using 3D point clouds. We review standard methods and recent advances in the use of machine and deep learning for change detection. In addition, the chapter presents a summary of 3D point cloud benchmark datasets from different sensors (aerial, mobile, and static), together with associated information. We also investigate representative evaluation metrics for this task. To finish, we present open questions and research perspectives. By reviewing the relevant papers in the field, we highlight the potential of bi- and multi-temporal point clouds for better monitoring analysis for various applications.

Keywords: 3D change detection; 3D point clouds; deep learning; machine learning; datasets

1.1 Introduction

The rapid development of 3D data acquisition is making the collection of massive point clouds faster than ever before, making them the future core topographic data for several applications [9]. However, in a dynamic world where everything is continually changing, data must be updated and used to detect and characterize the changes that have occurred. Change detection (CD) is defined as the process of recognizing the dynamics and changes in the earth's surface that occur between two or more epochs over the same geographic area [10].

CD has been widely studied in remote sensing applications, and many approaches based on two-dimensional (2D) images have already been proposed [11], [12], [13], [14], [15], [16], [17], [18], [19]. The availability of continuous large volumes of satellite images with short revisit times has created favorable conditions for the detection of long-term changes with improved temporal resolution. Similarly, 3D change detection (3D CD) is attracting more and more attention, due to the increasing availability of 3D data at different scales (district, city, region, and country). In real-world cases, change can range from rapid change (for example, in the context of autonomous driving [20], [21]) to slow change (e.g., remote sensing applications [15], [22], [23]) depending on the studied phenomenon, its frequency, magnitude and velocity. We divide change into two main categories: 3D tracking of homologous parts of a surface to compute a displacement field (fast change) and computing the distance between two point clouds when the homologous parts cannot be defined (slow change). Each category includes several methods to measure and track object dynamics. In 3D remote sensing applications, there has been increasing demand for 3D CD in the following fields: land use and land cover change detection [24], urban monitoring [25], [26], [27], forest changes [28], [29], [30], crisis monitoring [31], 3D geographic information updating [32], [33], [34], landslide and erosion monitoring [35], [36], [37], [38], [39], construction progress monitoring [40], and resources surveying [41], [42].

When dealing with multi-temporal point clouds, we talk about a point cloud with reference to a specific time, “epoch”. Due to the change in acquisition parameters from one epoch to another, it is not possible to make a direct comparison between two point clouds, because the sampling of points on the surface is not the same. Consequently, a point does not have its direct homologue on the second epoch, and, hence, no homology is possible between points; therefore, no direct calculation of point-to-point distance is possible. This aspect must be considered for 3D CD, either by preprocessing, if the density is different, or by using adapted methods (e.g., Cloud-to-Cloud) [43]. Change detection encompasses, in addition to displacement calculation and “from–to” change classification, other aspects, such as binary changes, multiclass changes, direction and magnitude of change, probability of change, temporal change trajectories (trends) and deformation analysis or abnormal behavior [22].

The 3D CD methods can be subdivided into Point-Based (PBCD), Object-Based (OBCD) [44], [45], [46], and Voxel-Based (VBCD) [47], [48], [49]. OBCD allows the detection of changes at an object level (segments or clusters that group a set of homogeneous points, or an instance that belongs to a known object class, like tree, car, building, etc.). The CD

step is largely influenced by the quality of the detection and segmentation of objects. VBCD methods rely on the discretization of space into grids, octrees, or voxels (e.g., occupancy grids), which are mainly used in robotic and indoor mapping applications [50], [51]. In this review, we focus mainly on the first category (PBCD), based directly on points with referrals to other approaches, without going into excessive detail. Although PBCD is more popular, due to its simple algorithms and relatively better quantitative results, applying these methods to multimodal point clouds often produces incorrect results. CD of large and outdoor scene point cloud faces many challenges, including incomplete data, noise, artifacts caused by temporary or moving objects, and cross-source point clouds captured by different sensor types. To address these challenges, many studies have proposed different methods to use voxels and objects as basic units for change detection [46], [47], [48].

Another aspect that needs to be introduced is semantic segmentation. Segmentation of 3D point clouds is the process of classifying point clouds into several homogeneous regions, where points within a region have the same properties. Segmentation is challenging, due to the high redundancy, uneven sampling density, and lack of explicit structure of point cloud data [52]. With the development of deep learning for point cloud semantic segmentation [53], [54], [55], high-level point clouds with semantic information can be obtained at unprecedented scales. Inspired by this development, recent research tends to incorporate high-level semantic features into point clouds CD to solve the problems of classic CD pipelines, such as binary CD (change, no change) and missing information about change type. However, finding changes is only one aspect of the problem. A subsequent crucial task is to make sense of them. What causes changes? Why and when do changes occur? What impact could changes have on other objects? Therefore, the simple question of what can be considered a “change” in point clouds is not trivial.

In this chapter, we offer a comprehensive review of change detection using 3D point clouds. We review the 3D CD methods used in remote sensing applications without integrating the 3D scene flow [56], [57], [58] or 3D object tracking [59], [60] methods. We introduce distance-based methods and learning-based methods with a focus on deep learning-based ones. The main contributions of our chapter are fourfold:

- Challenges related to the use of point clouds in CD and survey of 3D CD methods.
- Comprehensive review of the popular point clouds datasets used for 3D CD benchmarks.
- Detailed description of evaluation metrics used to quantify change detection performance.
- List of the remaining challenges and future research directions to advance the development of CD using 3D point clouds.

We structure the rest of the review as follows. Section 2 reviews the challenges related to 3D CD and relevant works on point clouds CD. Section 3 presents a summary of existing 3D point cloud datasets and evaluation metrics. Section 4 proposes a list of the remaining challenges for future research. Section 5 closes the chapter.

1.2 3D change detection using 3D point clouds

Our approach for compiling different works on 3D change detection using point clouds is based on previous reviews of the literature, as in [22], [61]. This gave us a quick and easy way to situate the state of the art, as well as to understand the proposed taxonomies to categorize the existing approaches. To the best of our knowledge, based on the study of literature, this is the first review dedicated to change detection using 3D point clouds to incorporate the latest machine and deep learning methods.

The articles reviewed in this chapter were published between 2004 and 2024. Our contribution is mainly focused on approaches using 3D point cloud data. For the rest of this chapter, we refer to change detection (CD) as the following setting: two point clouds or more, acquired at different times, covering the same area of interest, in which parts, objects, or surfaces move, change scale or color, distort, appear, or disappear between two different times. We assume that point clouds are registered (aligned in the same frame reference), can have different sources (dense image matching, laser scanning, etc.) and be acquired from any platform (aerial or terrestrial, etc.). Unlike 3D CD, in 2D CD in remote sensing, a multitude of state of the art with most recent methods already exist [19], [62], [63]. Therefore, before reviewing the methods found in the literature, we discuss the issues related to the use of 3D point clouds, the challenges associated with them, and their advantages and disadvantages over 2D data.

1.2.1 Challenges and specificities

Acquisition Challenges

The application of change detection on 3D point clouds presents many challenges. Indeed, changes in sensing conditions and unconstrained environments have a significant impact on the appearance of objects. Objects detected in different scenes or instances exhibit a range of variations. Even for the same scene, parameters such as scan timing, location, weather condition, sensor type, sensing distance, background, etc., produce significant variations for intra- and inter-class changes in 3D point clouds. We subdivide the major challenges related to the use of point clouds for CD into two key components:

Scan-related artifacts. All sensors are noisy, and the scanning system itself presents several artifacts which can be significant for applications of CD. One major source of artifacts on a scan is regions with non-diffuse reflection properties that refer to areas where pulses have been transmitted, but due to pulse absorption or specularity, not enough energy (if any) is returned to trigger a distance measurement (e.g., windows, water, and low reflectance surfaces). When data from another scan exists in these “hole” regions, it is considered a change while it is not. There is also, occasionally, noise caused by the presence of particles in the air when scanning, as well as unwanted points caused by the reflection of the laser pulse on a surface, such as a lake or a river. Hence, it is important to conduct a pre-processing step to clean up these artifacts before performing the change detection [64].

Occlusion. Due to occlusions, point cloud data are often incomplete. The occlusion of objects, or parts of objects, offers another variety of “interference”, with the probable

consequence of false CD. Points that appear in one scan but not in the other are considered potential candidates for natural change. To handle occlusion effects in point clouds, Ref. [65] proposed a new strategy for detecting "changed", "unchanged", and "unknown" buildings, where the latter class is applied to places where, due to lack of data in at least one of the epochs, it is not possible to reliably detect structural changes. Other research papers address this problem by using deep learning to fill in the occluded parts [66], [67], [68], [69], [70], [71], [72].

Three Dimensional Point Clouds Specificities

For any type of data to be processed, either in 2D or 3D, there are specific characteristics to consider. So, in addition to the point clouds challenges associated with acquisition, storage and manipulation phase, there are other challenges associated with the processing stage. We detail four challenges specific to point clouds that must be considered for each application, or algorithm, and, thus, be considered for CD.

Irregularity. Due to several acquisition parameters (e.g., the distance of the point from the sensor), point cloud data are irregular. This means that points are not homogeneously scattered in different regions of an object or scene, so some regions may have dense regions and others sparse ones, resulting in a changing density inside the same point cloud, as we can see in **Figure 4a**. This irregularity can be reduced by subsampling techniques using octree or spatial distance but cannot be eliminated [41].

Unstructured. Unlike images, point clouds in their raw form are not placed on a regular grid. Each point is scanned independently, and its distance to adjacent points is not always fixed, as shown in **Figure 4b**, which makes their spatial structuration complex. In analogy to images where pixels are represented on a two-dimensional grid, and the space between two adjacent pixels is always fixed, the point cloud is sometimes transformed into voxels to facilitate processing [47], [73].

Unorderedness. A point cloud of an object or a scene is a set of points derived from the object's surface (represented by XYZ coordinates, optionally with additional attributes). They are usually stored as a list in a file. As a set, the order in which the points are stored does not change the represented scene, so we say it is a permutation or order invariant. The unordered nature of point sets is illustrated in **Figure 4c**.

Rigid transformation. There are various rigid transformations in point clouds, such as 3D rotations, scale, and translations. However, these transformations should not affect the results and performance when using processing algorithms and especially deep neural networks.

To date, change detection in remote sensing has been primarily image-based, typically using object–background separation, simple subtraction between images, and, more recently, learning-based methods. These approaches have the disadvantage of imposing rigid constraints, such as static camera, the need to have the same viewpoint of the satellite, the need for recognizable landmarks, and sensitivity to shadowing and local illumination problems.

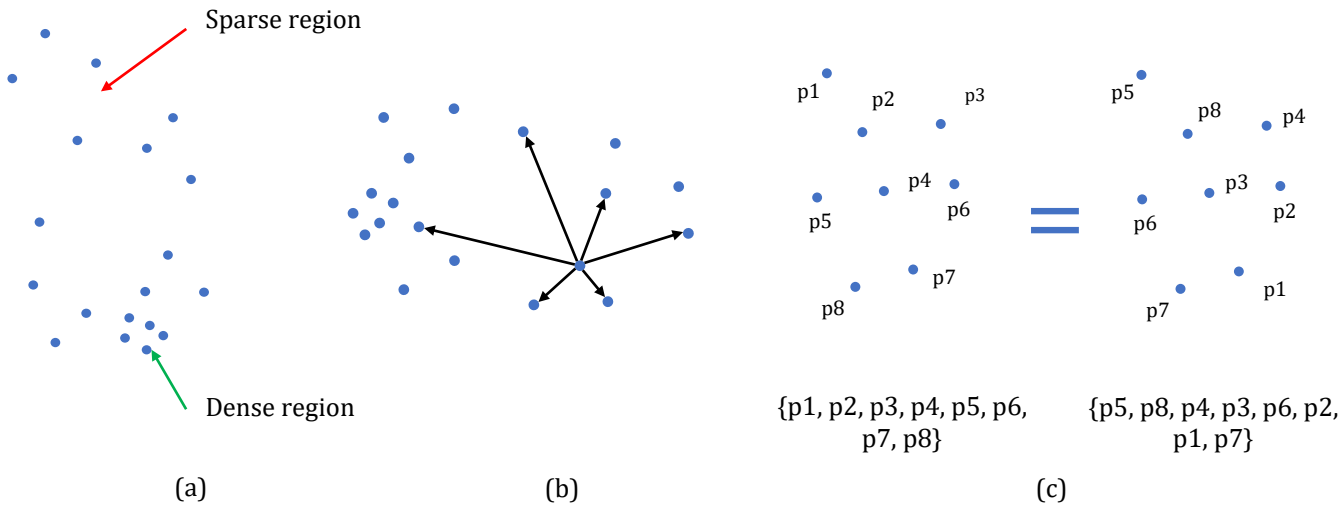


Figure 4. Challenges related to 3D point cloud processing. (a) Irregular: with dense and sparse regions. (b) Unstructured/no grid: each point is independent and the distance between adjacent points is not fixed. (c) Unordered: as a set, point clouds are permutation invariant.

The advantages and disadvantages of 2D data for CD have been known for a long time, while 3D CD work has emerged recently, mainly due to the availability of multi-temporal point clouds and the development of learning-based algorithms (**Table 1**).

Table 1. Summary of main differences between 2D and 3D CD specificities [22].

	2D CD	**3D CD**
Data source	Optical images, multi spectral images, RADAR images [74], Digital Surface, Terrain, and Canopy Models, and 2D Vector data.	Point clouds, InSAR (Interferometric SAR), Digital surface model, stereo and multi-view images, 3D models, building information models, and RGB-D images.
Advantages	Well-investigated [15], [75], [76], [77], available datasets [78], [79], [80], [81], [82], [83], available implementation [76], [82], [84].	Height component, Robust to illumination differences, Free of perspective effect, and provide volumetric differences.
Disadvantages	Strongly affected by illumination and atmospheric conditions. Limited by viewpoint and perspective distortions.	Unreliable 3D information may result in artifacts. Limited data availability. Expensive processing.

Despite the issues associated with using point clouds as a type of 3D data, they have several advantages which motivate their use for change detection. The main advantages of using 3D data over 2D data for change detection are summarized as follows: (1) Insensitive to illumination difference. As already stated, point clouds refer to spatial measurements of 3D objects; therefore, the comparison of the geometry of multi-

temporal data is independent of illumination conditions; (2) Insensitive to perspective distortions. Using point clouds, geometry comparison can be performed in a real three dimensional space, or any projected 2D space (subspace of the 3D space). In this case, they are not influenced by the point of view, as for 2D images where this effect is very noticeable; (3) Volume information. Change detection in 3D provides information on volumetric changes which paves the way for more applications, such as volumetric loss of forests, precise monitoring of construction progress, etc.

1.2.2 Data preprocessing

In remote sensing, image preprocessing is an essential first step, it includes geometric rectification and images registration, radiometric normalization, cloud and cloud shadow detection, atmospheric and topographic correction [3], [85]. For 3D point clouds, preprocessing is an essential step before applying different change detection algorithms. The purpose is to minimize changes due to characteristics we are not interested in, and to identify changes in which we are interested. So, we make the data at different epochs comparable. It includes removal of outliers, filtering, registration, and rasterization of DSMs (Digital Surface Models), DEMs (Digital Elevation Models) and nDSMs (normalized DSMs). We detail each step as follows:

Removal of outliers. The first step in preprocessing is the removal of outliers (unwanted points). They can be moving objects (a person, a car, etc.), vegetation, acquisition artifacts, noise, or points outside the area of interest. By removing these undesirable points, errors are reduced in the subsequent steps, such as normal calculation, registration, rasterization and change detection. The removal can be done manually by the operator or automatically, based on adapted algorithms [86], [87], [88]. In [36], the outlier removal algorithm calculates, for each point, the distance to all its neighbors and removes points having distances outside the point cloud's global mean and standard deviations.

Filtering. The purpose of filtering is to separate ground points from non-ground points. Several methods exist for this purpose. Ref. [89] classifies them into three categories, namely, mathematical morphology-based filters [90], [91], [92], [93], surface-based filters [94], [95], [96], and slope-based filters [89], [97], [98]. A further category can be considered which is segmentation-based filters [99], [100], [101], [102].

Registration. Multi-temporal point clouds need to be aligned before performing changes analysis [36]. They do not need to be acquired from the same view, unlike images, but must be georeferenced in the same coordinate system. For a point cloud acquired from a single station/viewpoint, if the sensor moves between epochs, if different sensors are used at different times or if at least parts of the surfaces deform between measurements, the same points are not measured or cannot be easily identified in the point clouds. Several algorithms exist for this task, comprehensive reviews of which can be found in [103], [104], [105].

Rasterization. For methods based on DSM, DEM or nDSM (see **Figure 5** for the difference) for 3D CD or multimodal methods which take different types of data as input, rasterization allows derivation of a DSM from the point cloud. First, the point cloud is structured into a grid (e.g., cell size 1 m). Then, the lowest point of each cell of the grid is

selected for triangulation as described in [106]. A similar method can be used for the rasterization of DEMs, but only the ground points are used in the triangulation. The nDSMs are derived through subtraction of the appropriate DEM from the DSM, as specified in the next formula:

$$nDSM = DSM - DEM \quad (1)$$

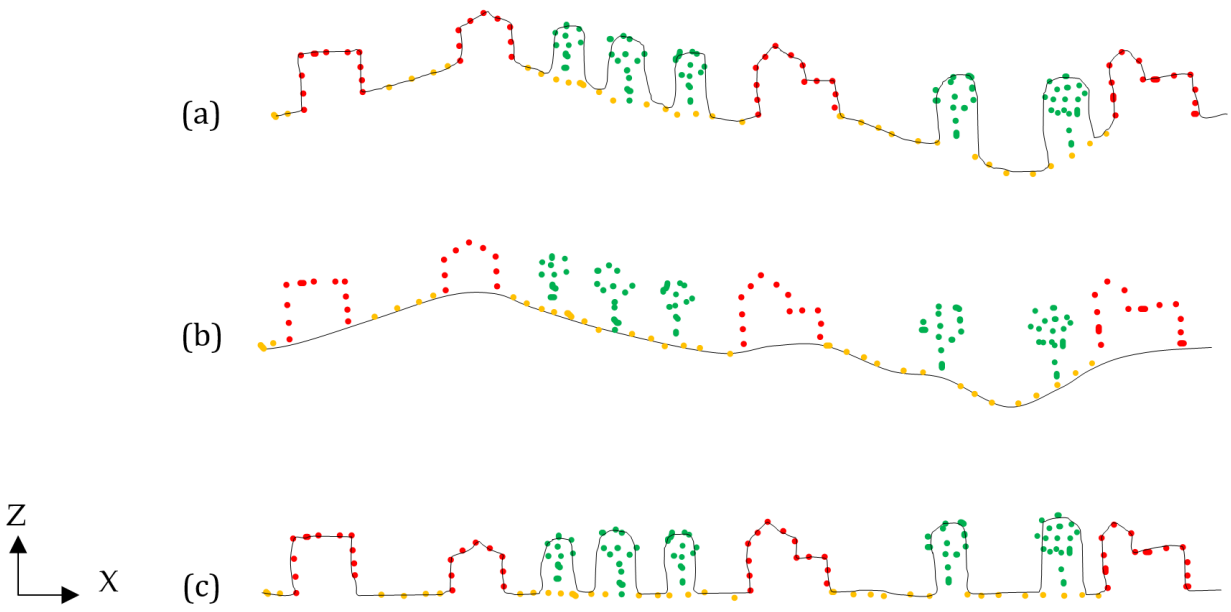


Figure 5. Schematic diagram to show the difference between: (a) DSM (Digital Surface Model), (b) DEM (Digital Elevation Model), and (c) The nDSM (normalized Digital Surface Model).

1.2.3 Three dimensional change detection methods

It is important to note that change detection techniques can be applied to a variety of input data (3D point clouds, meshes, 2D images or a combination of both). In our chapter, we focus on 3D point clouds. Based on our review, we find that the categorization of these techniques can be done in different ways based on several criteria:

- Change Unit. This taxonomy depends on the basic unit used in the CD process, such as methods based on points, voxels, objects and rays;
- Order of classification and change detection. In this categorization, difference exists between methods that proceed to change detection and then classification (pre-classification methods), those that proceed to classification first (post-classification methods), and those that integrate the two steps into one (integrated);
- Used technique. Methods are classified here based on the technique used whether it is based on distance or learning, etc;
- Target. This depends on the application domain: urban, forestry, maritime, etc.

Many different point cloud-based change detection techniques have been proposed in different contexts of remote sensing, self-driving vehicles or robotic applications. In general, many methods use point clouds to compare 3D representations of the environment in different states. Below (in **Table 2**), we summarize the methods found in the literature and classify them according to the used input data (LiDAR, images which can be remapped into orthophotos or the originals ones, and maps), the order of classification and change detection (integrated, pre- and post-classification change detection) and the target context (building, tree, vegetation, etc.). In addition to these criteria, we indicate methods that transform the point clouds into a DSM (Digital Surface Model) to work on it. Most of the work tend to use point clouds more often and integrate classification and change detection in one step to overcome the limitations related to the other types. We can also see that most of the studies concern the urban environment and, more specifically, buildings.

Table 2. Overview of 3D CD methods by input data, approach, and change detection class.

Authors	Year	Input Data			Change Detection Approach	Change Detection Class
		LiDAR	Image	Maps		
Matikainen et al. [107]	2004	X	Ortho	X	Post-classification	Building
Vu et al. [108]	2004	X			Pre-classification DSM-based	Building
Vosselman et al. [109]	2004	X		X	Post-classification	Building
Choi et al. [110]	2009	X			Post-classification	Ground, vegetation, Building
Matikainen et al. [111]	2010	X	Ortho	X	Post-classification	Building
Stal et al. [112]	2013	X	Ortho		Post-classification	Building
Malpica et al. [113]	2013	X	Original		Post-classification	Building
Teo et al. [114]	2013	X			Post-classification DSM-based	Building
Pang et al. [64]	2014	X			Pre-classification DSM-based	Building
Zhang et al. [115]	2014	X			Pre-classification	Ground
Tang et al. [116]	2015	X		X	Post-classification	Building
Awrangjeb et al. [34]	2015	X		X	Post-classification	Building
Xu et al. [65], [117]	2013, 2015	X			Post-classification	Building
Xu et al. [118]	2015	X			Pre-classification	Building, tree
Du et al. [119]	2016	X	Original		Pre-classification	Building
Matikainen et al. [120]	2016	X	Ortho	X	Post-classification	Building
Matikainen et al. [121]	2017	X	Ortho	X	Post-classification	Building, roads
Kaiguang et al. [122]	2018	X			Post-classification	Forest
Marinelli et al. [123]	2018	X			Post-classification	Forest
Zhang et al. [124]	2019	X	Ortho		Integrated	Building
Zhang et al. [125]	2019	X	Ortho		Integrated	Building
Yrttimaa et al. [30]	2020	X			Post-classification	Forest
Fekete et al. [126]	2021	X			Post-classification DSM-based	Tree
Huang et al. [127]	2021	X	Original		Post-classification	Building

Ku et al. [128]	2021	X	Integrated	Building, street, tree
Iris et al. [129]	2021	X	Integrated	Building
Tran et al. [130]	2021	X	Integrated	Ground, vegetation, building
Zhang [131]	2022	X	Integrated	Building
Dai et al. [44]	2022	X	Integrated	Building
Irsi et al. [132]	2023	X	Integrated	Urban
Zhan et al. [133]	2024	X	Integrated	Ubran

The new categorization we propose is based on the algorithms used for change detection, regardless of the basic unit, the context studied or the order of classification and change detection. It classifies change detection methods into three main types that can, in turn, incorporate subtypes. The first type includes standard methods (also called distance-based methods), the second type includes machine learning-based methods that use handcrafted features, and the last type includes the deep learning methods that extract more abstract features without user specification (see **Figure 6**).

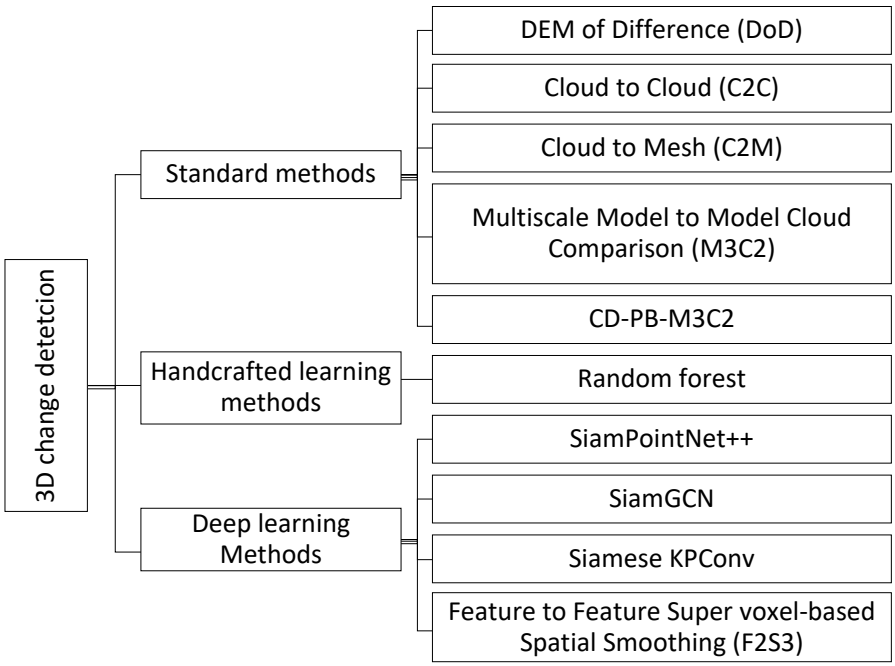


Figure 6. The taxonomy of 3D CD approaches with the methods we will detail in the next sections.

Standard Methods

This category of 3D CD includes point clouds comparison methods based on the calculation of the distance between two points. They establish displacement based on proximity in Euclidean space. There are approaches based on the difference between digital elevation models (DEMs), the distance between point clouds using the nearest distance, and multi-scale point clouds comparison. We detail each of these as follows:

DEM of difference (DoD): This has a simple concept and easy implementation; it quantifies the surface change based on the DEM derived from a 3D point cloud. One of

the most common methods in this category is the difference between DEMs that estimate elevation change on a cell-by-cell basis where the change is derived along a single, predefined direction (Z-axis) [134], [135], [136]. This simplicity reveals a limitation in complex contexts, such as overhangs and near vertical slopes, where the vertical difference is not sufficient.

Cloud-to-cloud comparison (C2C): Point cloud comparison is the simplest and most effective method for deformation and CD [43]. The change is detected by calculating the Euclidean distance between individual points in a reference point cloud (epoch A) and the respective nearest neighbor point (NN) in the target point cloud (epoch B). We illustrate this process in **Figure 7a**. Its simplicity is its major advantage, since it does not require the calculation of normal, and, therefore, no local modeling of the surface is needed (e.g., triangulation or plane adjustment).

Cloud-to-Mesh comparison (C2M): Like the C2C method, the C2M method computes displacements using the nearest Euclidean distance between each point in a source point cloud and the nearest facet (or to an edge, if the orthogonal projection of the point does not fall on any facet) in the target mesh of the triangulated point cloud [137], [138]. The principle of this method is illustrated in **Figure 7b**. Its main limitation is its need to mesh the target point cloud. This can generate triangular surfaces with holes and artifacts, which leads to false displacement and change detection.

Multiscale Model to Model Cloud Comparison (M3C2): This method estimates the displacement based on the points of interest found in each point cloud (e.g., sub-sampled points of the point cloud or even the entire source point cloud) [139]. It has two main steps: (1) Surface normal estimation and orientation in 3D space at a scale consistent with the locale surface roughness; (2) Measurement of the mean surface change along the normal direction with the explicit calculation of local confidence interval (σ). Its main advantage is that it works directly on point clouds without needing to mesh or grid them. It estimates a confidence interval for each distance measurement depending on the roughness and registration error of the point clouds (see **Figure 7c**). However, the M3C2 performs less well when the changes occur in different directions than the direction of change computation, and also when the level of detection (see Section 3.2) exceeds the magnitude change.

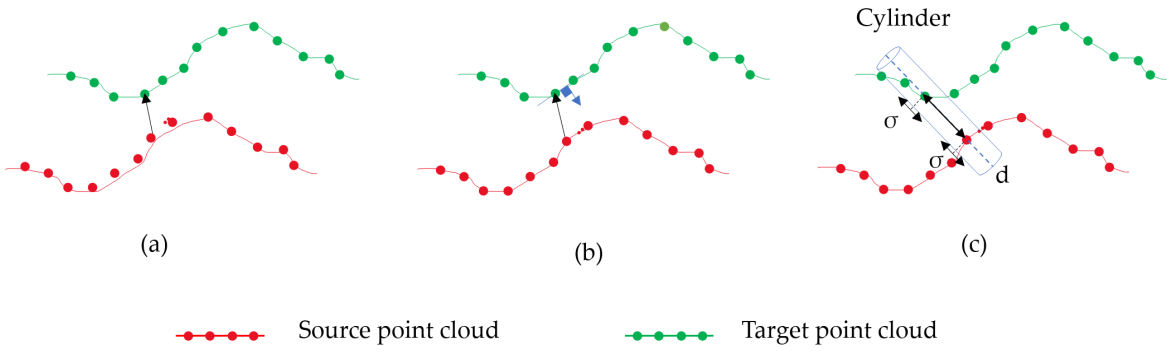


Figure 7. Existing approaches of point-to-point distance calculation. (a) Cloud to Cloud. (b) Cloud to Surface. (c) Multiscale Model to Model Cloud Comparison.

There are two problems associated with this type of method, which relies on the point-to-point distance calculation for CD. The first issue is related to density, which is variable between two epochs and within the same epoch. This is affected not only by the distance of the point from the acquisition sensor but also by the change in the type of sensor and the acquisition mode. The second problem is related to the inefficiency of these methods to deal with occluded areas in point clouds, because they do not consider free spaces. To counter these two problems, there are ray-based methods. These methods require that the sensor positions be known for each instant to recreate a bundle of rays representing the pulse path and the measured point [61]. However, this type of method is highly point-of-view dependent and cannot be generalized to data without information about the sensor positions.

A further challenge in 3D CD is the quantification of small changes with low uncertainty. A recent paper proposed an improved version of M3C2, called Correspondence-Driven Plane-Based M3C2 (CD-PB M3C2) [140]. Based on two points clouds at epoch A and epoch B, this method uses a three steps workflow (**Figure 8a**). The first is to extract the planar surface using a region-growing segmentation (**Figure 8b**). The second is the plane's correspondence search through a binary random forest classifier (RF) (**Figure 8c**). The third is the quantification of change and uncertainty (level of detection) through the calculation of M3C2 distance between each plane and its corresponding one (**Figure 8d**). This approach, based on plane correspondence, gives results seven-fold better than the M3C2, in terms of uncertainty associated with topographic change, and shows high performance for quantifying small-magnitude (less than 0.1 m) changes. The other main advantage of the CD-PB-M3C2 is that, by using the matching planes, it is not necessary to determine the direction of change a priori, as the feature similarities are used regardless of the absolute position of the planes. Nevertheless, the use of planes constitutes its own disadvantage because it does not allow high-level recognition of objects other than planes.

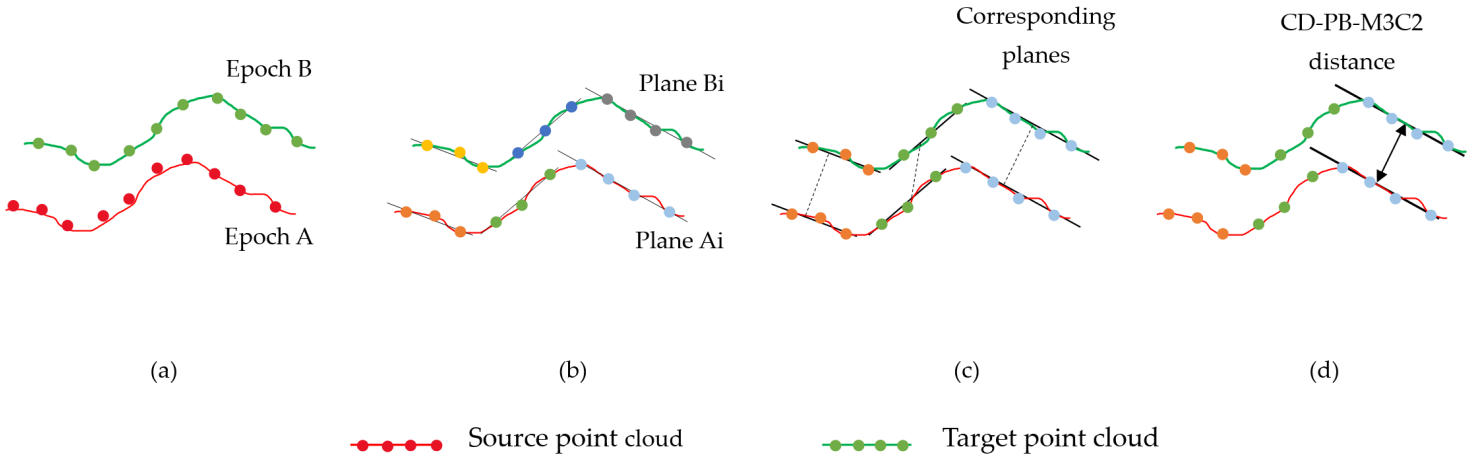


Figure 8. Correspondence-driven plane-based M3C2 as described in [140]. (a) Input point clouds. (b) Extraction of planar surfaces. (c) Plane correspondence search. (d) Quantification of change and uncertainty.

To summarize, the problem with traditional methods for CD and displacement analysis is establishing the correspondences between point clouds in the Euclidian space. However, these correspondences just represent the distance between the surfaces, not the actual displacement of points on the surfaces. Therefore, previous methods (C2C, C2M, M3C2 and CD-PB-M3C2) struggle to estimate the correct and significant change in a specific case, mainly in the case of parallel motion to the surface. To resolve this problem, several methods have been proposed in the literature, such as the fusion of point clouds and RGB (Red, Green, Blue) images, as in [141]. The latter estimates 3D change and displacement vectors using the corresponding points obtained, based on 2D RGB-Depth image in features space, as illustrated in **Figure 9**. The matching is not distance-based but uses the correspondence of the extracted features around each point in a radius r to find the corresponding point in the target point cloud. With the same consideration, recent machine and deep learning methods propose to establish correspondences in the feature rather than the Euclidean space only.

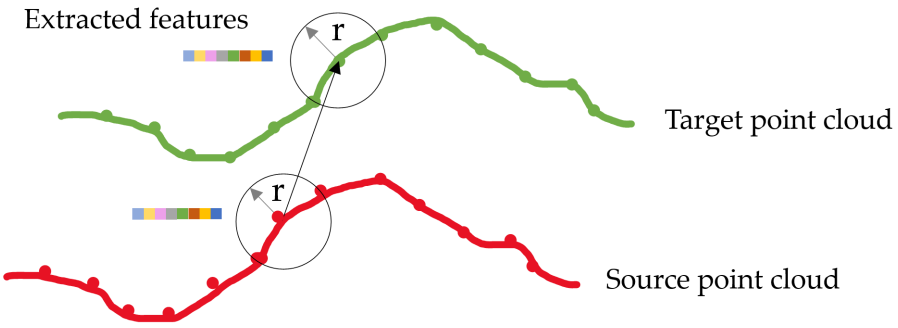


Figure 9. Correspondence between points at different epochs in features space.

Machine Learning with Handcrafted Features

In this section, we show works using handcrafted features in machine learning algorithms. These recent approaches for CD using 3D point clouds integrate the classification and change detection at the same time using handcrafted features [27], [130]. The general idea of these methods is to extract inter- and intra-epoch features and then classify them using a learning model to obtain change classification results, as illustrated in **Figure 10**.

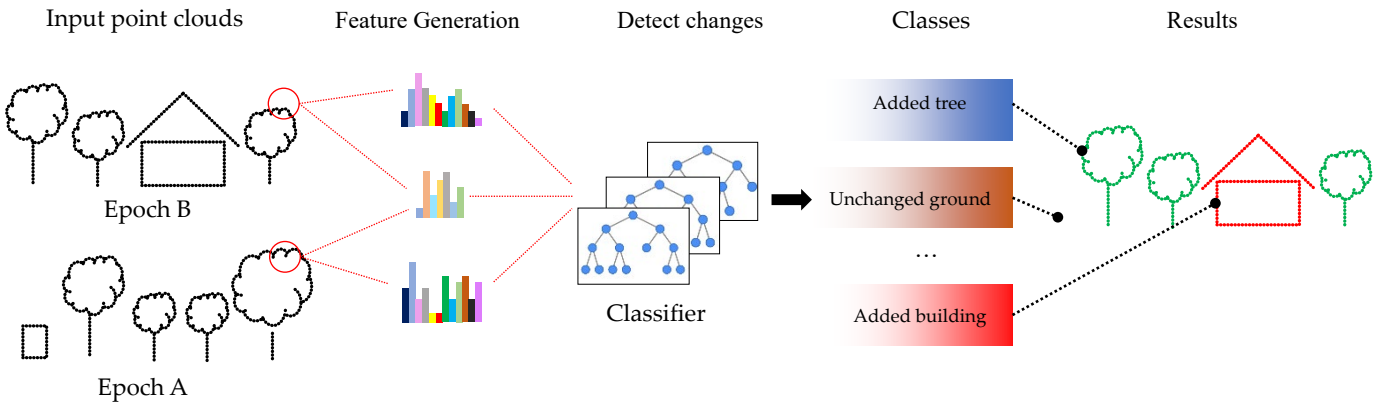


Figure 10. General 3D CD framework using handcrafted features.

For more details, we present the workflow proposed by [130]. Point clouds at two different epochs (A and B) are merged to extract four types of features: features describing the point distribution (e.g., planarity, verticality, linearity and omnivariance), feature height above the terrain, features specific to the multi-target capability of laser scanning (e.g., return number and number of returns) and features combining point clouds from both epochs to identify change. The proposed methods take as input two cleaned and georeferenced point clouds in the same coordinate system (no registration needed). Then, the point clouds are merged to generate the four specified features, as shown in **Figure 11**, and, finally, provide change detection classification type using random forest classifier.

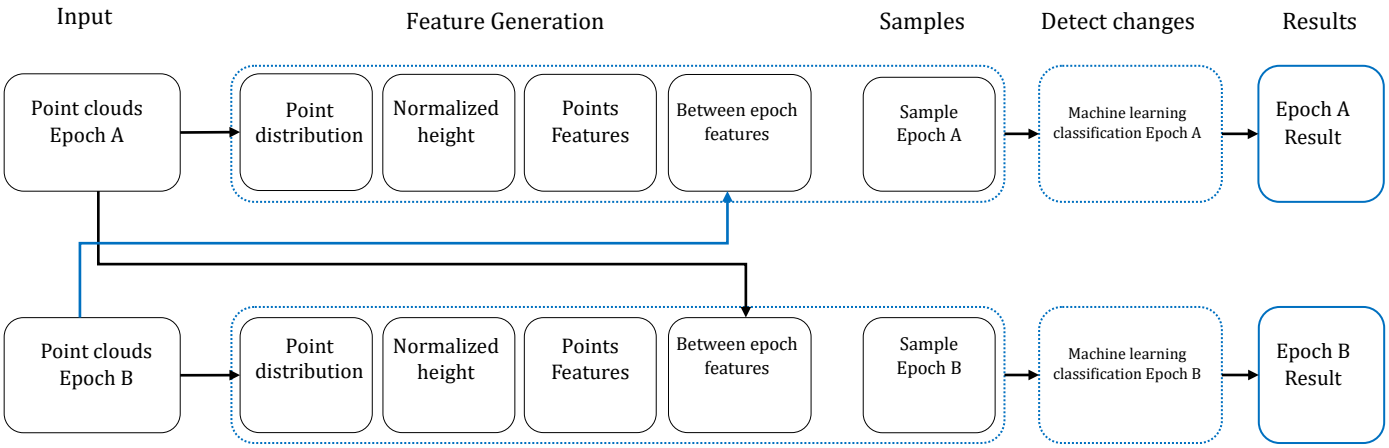


Figure 11. Integrated CD and classification of point clouds using handcrafted feature as in [130].

Despite its advantages of fusing classification and CD in a single step, as well as being able to detect change in multiple classes simultaneously (e.g., tree, building, soil, etc.), machine learning methods depend heavily on the initial training data. This problem can be partially solved by using unsupervised machine learning methods.

Deep Learning Methods

To overcome the problem of change detection based only on distance correspondence in a Euclidean space, deep learning methods propose to create more abstract features without the need for user specification. In [142], [143], the authors proposed a new deep learning framework for point clouds displacement and change analysis, called Feature to Feature Super voxel-based Spatial Smoothing (F2S3). It is divided into two mains step: (1) Estimation of an initial 3D displacement vector field by determination of point-to-point correspondence in the feature space, and (2) Filtering and smoothing of the initial 3D vector field pipeline (see **Figure 12**). The proposed concept of F2S3 is not to rely on proximity in Euclidean space (distance) but to create a correspondence between points at different times based on proximity in feature space. This proximity is covered by local feature descriptors, which describe the geometric information of the local neighborhoods of the point of interest (e.g., a sphere of radius *r*, as specified in **Figure 9**). Spectral and radiometric features (e.g., color, intensity, or multispectral bands) can be used in addition to geometric features, but these are usually neglected and not considered in the local feature descriptors, because of constraints related to

changing acquisition conditions, environment, sensors, etc. Thus, by establishing the corresponding points within the feature space, F2S3 is sensitive to displacements along the surface. It was demonstrated in this work that F2S3 outperformed the standard methods (C2C, C2M, M3C2) on real-world geo-monitoring datasets when the hyper-parameters were chosen appropriately [142], [143].

However, the available current implementation of F2S3 is computationally very complex, not fully automated, and requires in-depth knowledge of the algorithm and the deformation process to choose the right hyper-parameters. To solve such problems, Ref. [38] proposes a further step and integrates the F2S3 workflow into a fully automatic pipeline (**Figure 12**). They propose a tiling procedure to create smaller point tiles to facilitate their processing in an efficient and parallel way (to process large point clouds). They also propose to replace the hyper-parameters that require in-depth knowledge of the user with values derived directly from the input point clouds. To overcome the problem of processing time and memory complexity, they propose using more efficient local feature descriptors, as proposed in [144]. These learning-based methods have been shown to outperform all the traditional methods already mentioned in Section 2.3.1.

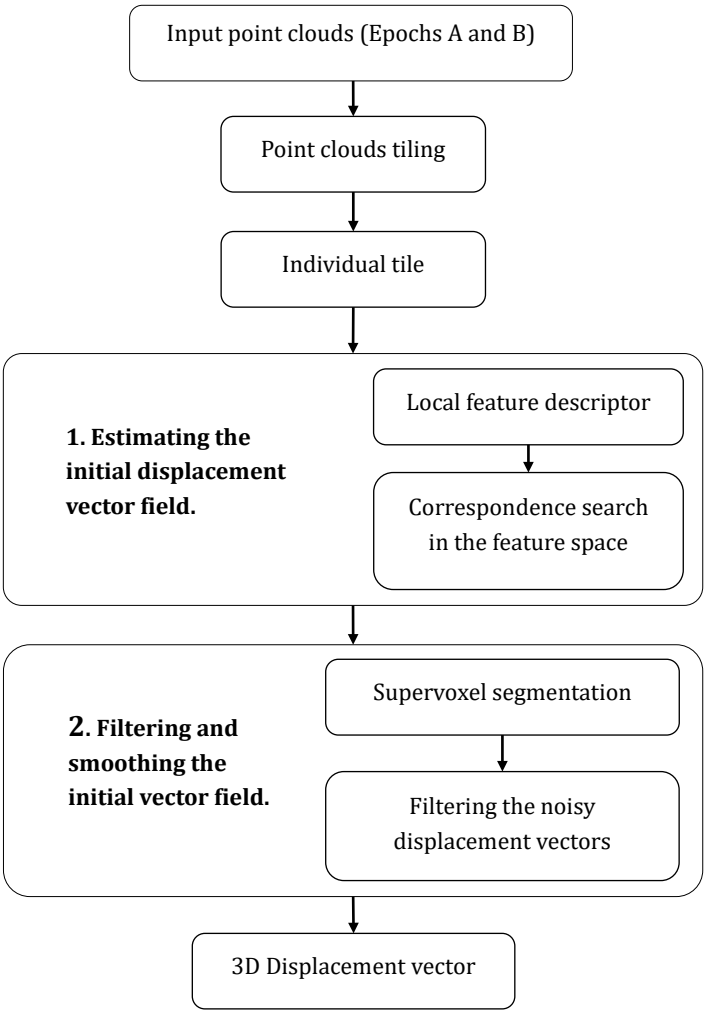


Figure 12. Workflow of the modified F2S3, as described in [38].

In another work, Ref. [128] proposed a Siamese Graph Convolutional Network (SiamGCN) for 3D point clouds CD. The edge convolution (EdgeConv) operator is adopted to extract representative features from point clouds (**Figure 13**). Then, a Siamese architecture, [128], based on the graph convolutional networks, is proposed to identify the change type of any two input point clouds (A and B) from two different epochs. The source code of their approach is publicly available at https://github.com/kutao207/SiamGCN (last accessed on 28 August 2022). The authors also evaluated three algorithms, including one handcrafted and two learning-based methods, on the 3D CD dataset (more details in Section 3.2). The first one is point clouds change detection with hierarchical histograms (named PoChaDeHH). The second one is a 3D PC CD for street scenes (named HGI-CD). The third is the SiamGCN (see **Figure 13**). Although the handcrafted algorithm can achieve relatively balanced results on the overall and per-class accuracy and mean intersection over union (mIoU), it was obvious that learning-based methods achieved overwhelming performance.

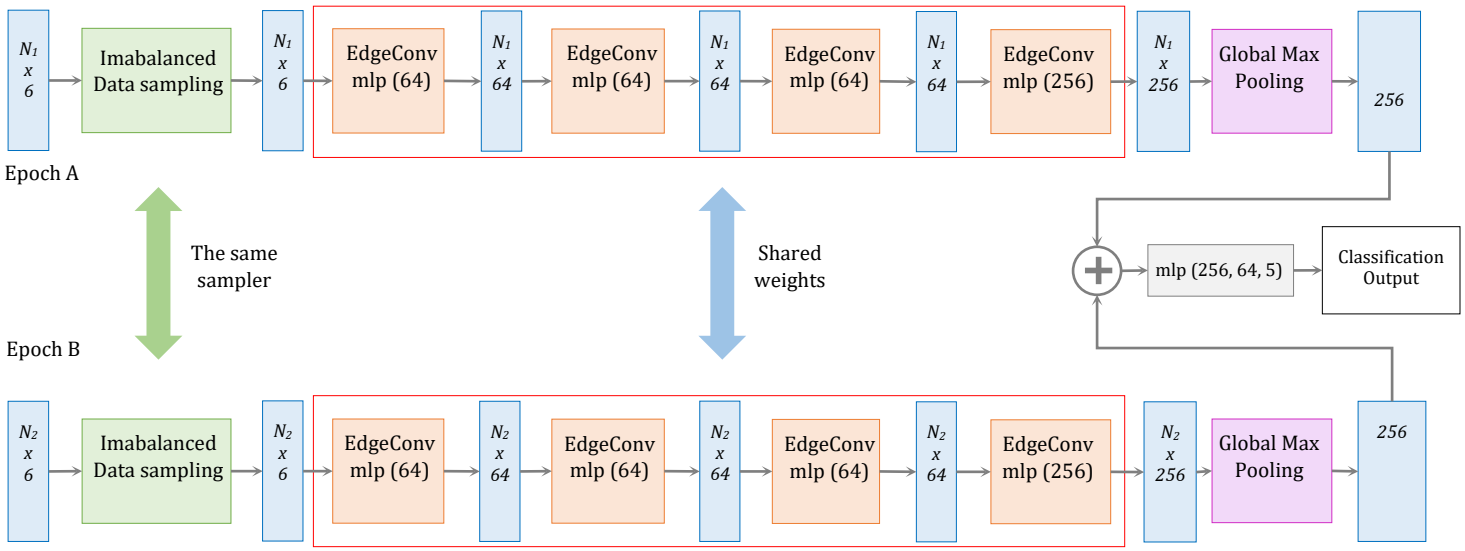


Figure 13. SiamGCN network architecture, as described in [128].

In [131], the authors proposed a method to detect building changes between LiDAR (Light Detection And Ranging) and photogrammetric point clouds. With consideration of the fact that semantic segmentation and CD are correlated, they suggested the SiamPointNet++ model that combine the two tasks in one framework (see **Figure 14**). The method outputs a pointwise joint label for each point. If a point is unchanged, it is assigned a semantic label (e.g., building) and if a point is changed, it is assigned a change label (new building). The semantic and change information is included in the joint labels with minimum information redundancy.

The combined Siamese network learns both intra-epoch and inter-epoch features. Intra-epoch features are extracted at multiple scales (sphere radius r) to embed the local and global information. Inter-epoch features are extracted by Conjugated Ball Sampling (CBS) and concatenated to make change inferences (**Figure 14**). For the decoder layers, the DIM feature vectors are interpolated to the raw LiDAR points locations, instead of

the raw Dense Image Matching (DIM) points locations. This ensures that the DIM features are calculated at the same centroids as the LiDAR data. Only feature vectors extracted at the same centroids can be compared. The authors point out that even if there is no DIM point in the conjugate ball of a LiDAR point, a pseudo feature map is calculated at the same centroid to "inform" the model that the neighborhood of the ball in the DIM data is empty [131].

Experiments conducted in a study area in Rotterdam, Netherlands, indicated that the network was effective in learning multi-task features. It was invariant to the permutation and noise of inputs and robust to the data difference between LiDAR and dense image matching data. Compared with a sophisticated object-based method and supervised change detection, this method requires much fewer hyper-parameters and less human intervention, but achieves superior performance, as stated by the authors.

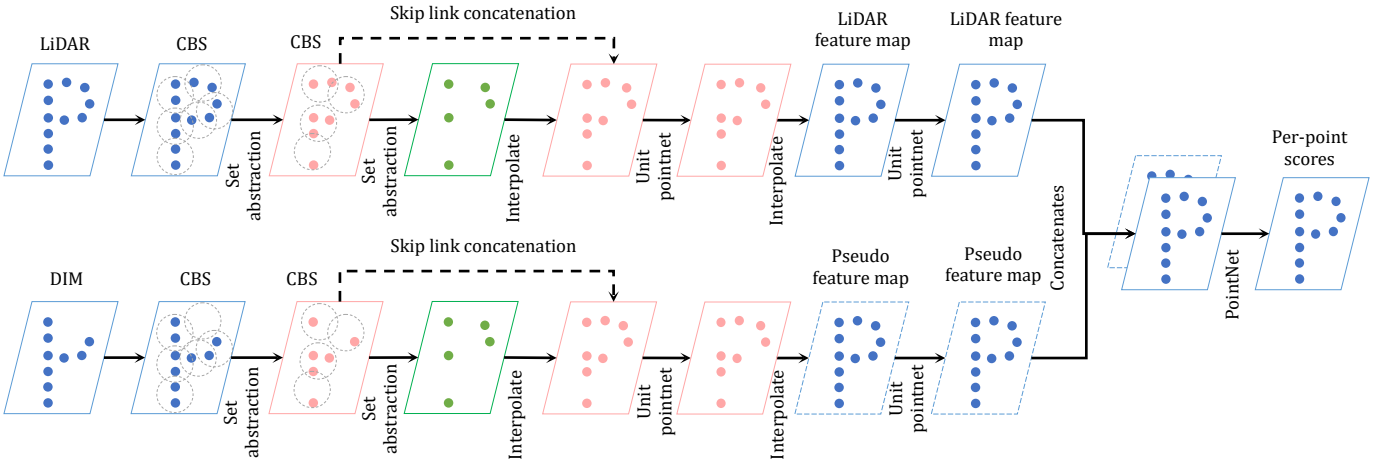


Figure 14. SiamPointNet++ network architecture [131].

The same author proposed in [124] a Convolutional Neural Networks (CNN) architecture for multimodal CD. Change in buildings was detected between a digital surface model (DSM) derived from point clouds, and a dense image matching point clouds using feed-forward CNN, which showed high performance for using multimodal data.

Using the same analogy of the Siamese architecture for 2D CD, Ref. [129] proposed to extend the Siamese to 3D point clouds. They proposed embedding the KPConv [145] architecture used for semantic segmentation into a deep Siamese network where both point clouds would pass through the same encoder with shared weights. Similar to the usual encoder–decoder architecture with skip connections, at each scale of the decoding part, they concatenated the difference of extracted features associated with the corresponding encoding scale, as shown in **Figure 15**. The authors compared the Siamese KPConv results to machine learning hand-crafted methods presented previously in [130]. The comparison results showed that the deep learning method outperformed the machine learning one for all evaluation metrics. An improvement of about twenty-seven points of IoU (Intersection over Union) over classes of change was observed.

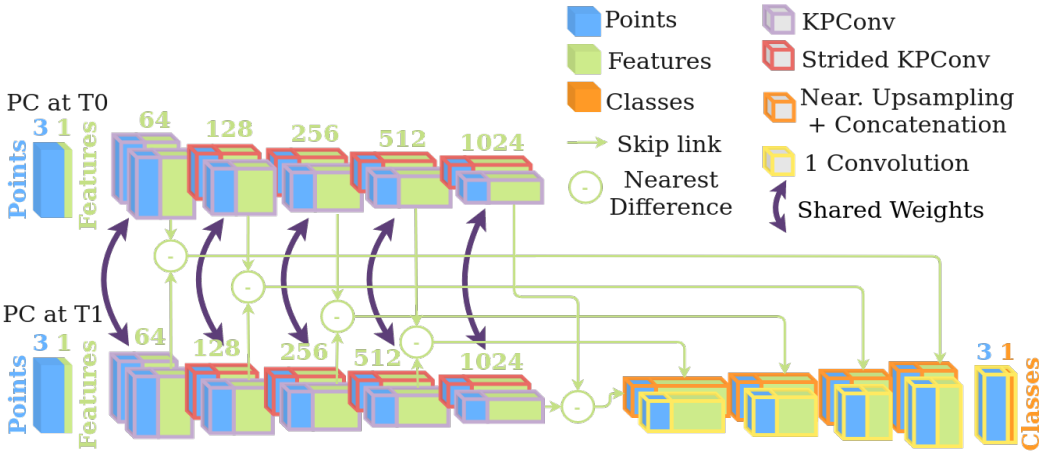


Figure 15. Siamese KPConv network architecture. Source: Image adopted from [129].

All conducted works in this section have shown that deep learning methods largely outperform traditional methods, either in terms of change classes number or in terms of evaluation metrics. Their main advantages lie in the ability to understand structured objects at a global scale, thus leading to correct classification of hidden parts, and so resolving the occlusion part in point clouds. However, the problem of occlusions and variability of density and distribution of points between the different epochs is not completely solved. As can be noticed, learning-based methods largely depend on the availability of training data to achieve high CD quality. In the following section, we present multi-temporal point clouds benchmarks which are accessible in Open Access, as well as the metrics used in the literature for performance comparison.

1.3 Benchmarks

A variety of methods are proposed to detect changes in literature, but the choice of the right one is not obvious. Therefore, it is essential to adopt appropriate metrics to use for performance evaluation in the same dataset. In this section, we list publicly available benchmark point clouds datasets and provide some standard evaluation metrics used to compare the performance of the algorithms, for application in future research.

1.3.1 Datasets for 3D change detection

This section presents different types of commonly used point clouds datasets for 3D CD. We classify the existing datasets related to our topic into two main types: unannotated datasets, which do not contain semantics information about objects classes or change type, and annotated datasets. The latter is the most useful because they pave the way for the rapid development of CD applications and exploitation of 3D data using machine learning methods and deep learning networks.

OpenTopography. This is a United States National Science portal that facilitates access to high-resolution data and related tools and resources [146], [147]. It provides open access to LiDAR and photogrammetry point clouds and photos with on-demand processing tools to generate derived products. The one we are interested in is the change map created from the available multi-temporal point clouds. One can download point clouds acquired at different locations in high resolution. It also offers the possibility to use two services on demand: the first one is vertical differencing, which

aims to measure landscape change by differencing DEMs (Digital Elevation Models) to see the topographic change from processes, including urban growth, flooding, landslides, wildfires, and earthquakes [148]. The second one is 3D differencing, which aims to detect horizontal and vertical change when the landscape shifts during earthquakes and landslides [148]. This dataset is not annotated into change classes, and it is accessible at: https://opentopography.org/ (last accessed on 28 August 2022).

AHN1, AHN2, AHN3, AHN4 and AHN5. The Actueel Hoogtebestand Nederland (AHN) is the digital height map for the whole of the Netherlands. What is interesting about this is that there is repeat LiDAR acquisition AHN1, AHN2, AHN3 and a recent AHN4 [149]. AHN1 (1997–2004) was initiated by Waterboards, Ministry of Infrastructure and Water Management and Provinces to manage the water systems and water security, then AHN2 (2007–2012) and AHN3 (2014–2019). There is an upcoming version, AHN4 (2020–2022), which is denser than ever before. These datasets already contain a pre-classification, but are not annotated in type of change, link https://www.ahn.nl/ (last accessed on 28 August 2022).

Abenberg—ALS test dataset. This dataset contains the point clouds acquisition of Adenberg, Germany (49.2416° N, 10.9636° E), using aerial laser scanner (ALS) RIEGL LMS-Q560 (version 2006). The first was acquired on 18 April 2008, by four ALS strips in a cross pattern, resulting in an accumulated point cloud which includes 5,400,000 points (**Figure 16**) with an average point density of 16 pts/m^2. The second was acquired on 31 August 2009, using the same sensors and a similar setting as specified by [150], resulting in a point cloud of 6,200,000 points with an average point density of 21 pts/m^2. In addition to the coordinates of 3D points (XYZ), the data sets contain the local normal directions, sensor positions, and results of pre-classification (ground, vegetation, and building). These multi-temporal multi-viewpoint clouds data are well suited for the development and evaluation of 3D CD methods in urban areas, and the investigation of other applications of ALS data, e.g., city modeling and city model updating.

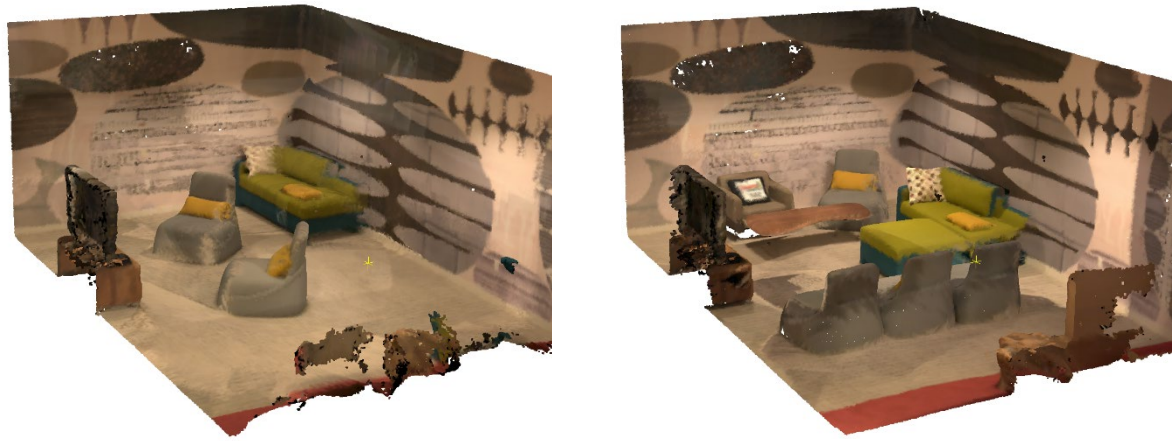
(**a**) (**b**)

Figure 16. (a) Adenberg data 2008 and (b) Adenberg data 2009.

4D objects by changes. This dataset contains an hourly dataset of Terrestrial Laser Scanning (TLS) point clouds acquired in the frame of coastal monitoring at the sandy

beach of Kijkduin (52°04′14″ N, 4°13′10″ E), Netherlands, over a period of five months [151], [152]. Link: https://doi.org/10.11588/data/4HJHAA (last access on 28 August 2022).

ICRA 2017—Change Detection Datasets. This contains three indoor datasets (living room, office, lounge) acquired using Google Tango tablets, mainly their RGB-Depth sensors with an operating range of 0.4 m to 4.0 m, and with a resolution of 320 × 180 pixels, as specified in [153]. The first one (living room, as in **Figure 17**) contains nine hand-held observations (9 epochs) in a controlled indoor environment. The second one (office) consists of four recordings of a controlled office environment recorded from the center of the room using a tripod. The third one consists of ten hand-held observations in an uncontrolled environment; an accessible meeting area observed over the course of two weeks. The link to the dataset is https://projects.asl.ethz.ch/datasets/icra2017changedetection (last accessed on 28 August 2022).



(**a**) (**b**)

Figure 17. Observations at two epochs (a and b) of the living room (controlled environment).

PLS dataset of Kijkduin beach–dune. A high-resolution 4D terrestrial laser scan dataset of the Kijkduin beach–dune system, Netherlands[154]. The beach was scanned hourly for 190 days. between 11 November 2016, and 26 May 2017, by a Permanent Laser Scanner (PLS) mounted in a permanent place at 38 m height above the mean sea level. The dataset is georeferenced and contains 4082 hourly point clouds, each one containing between one and ten million 3D points. It contains additional attributes, such as the laser return and intensity.

CG-PB-M3C2. This dataset includes six points clouds acquired bi-weekly by a TLS within the summer of 2019 in the lower tongue area of the rock glacier Außeres Hochebenkar, Austria. The dataset contains approximately 222 million points per epoch, with a point spacing ≤ 0.01–0.11 m (mean 0.03 m). More details on the point clouds and their spatial coverage can be found in [140] and at https://doi.org/10.11588/data/TGSVUI (last accessed on 28 August 2022).

Near-continuous 3D time series. This dataset contains data to perform spatiotemporal segmentation in time series of surface change data for synthetic data and hourly snow cover changes acquired by a terrestrial laser scanner (TLS) [155]. The link to the dataset is https://doi.org/10.11588/data/1L11SQ (last accessed on 28 August 2022).

Change3D Benchmark. The data is provided by CycloMedia. It consists of annotated "points of interest" in street-level colored point clouds gathered using vehicle-mounted LiDAR sensors, in 2016 and 2020, in the city of Schiedam, Netherlands [128]. This dataset focuses on street furniture, with most of the labels corresponding to traffic signs (**Figure 18**). Although other objects, such as advertisements, statues, and garbage bins, are also included. Labeling was done through manual inspection. The dataset proposed over 78 annotated street-scene 3D point cloud pairs. Each point cloud pair represents a street scene in two different years and contains a group of changed or unchanged objects. Each object pair is assigned one of the following labels: (1) No change, (2) Added, (3) Removed, (4) Change and (5) Color change. The link to the dataset is https://kutao207.github.io/ (last accessed on 28 August 2022).

(a) Single street scene with multiple changed objects.

(b) *color_change* example

(c) *change* and *added* example

Figure 18. Change3D dataset with labeled points and two examples of change in color and addition. Source: Image adopted from [128].

TUM City Campus—MLS test dataset. This dataset is situated in Munich, Germany (48.1493° N, 11.5685° E), and covers an area of about 29,000 m2. The first one was acquired on 18 April 2016, using Mobile Laser Scanning (MLS), resulting in more than 8000 scans (rotations of the scanner head) with 1.7 billion points [156], [157]. An Additional epoch (TUM-MLS-2018) was acquired on 19 December 2018, resulting in 10,500 scans (rotations of the scanner head) with 2.2 billion points. Parts of the two datasets (**Figure 19**) were labeled and contain semantic information (Artificial Terrain, Natural Terrain, High Vegetation, Low Vegetation, Building, Hardscape, Artifact, and Vehicle). The authors have added a less dense, old epoch (TUM-ALS-2009), acquired by airborne laser scanning in 2009, which offers further research possibilities. So, using these multi-temporals datasets, methods for 3D CD can be developed and tested. The datasets and annotations can be downloaded at: http://s.fhg.de/mls1 (last accessed on 28 August 2022).

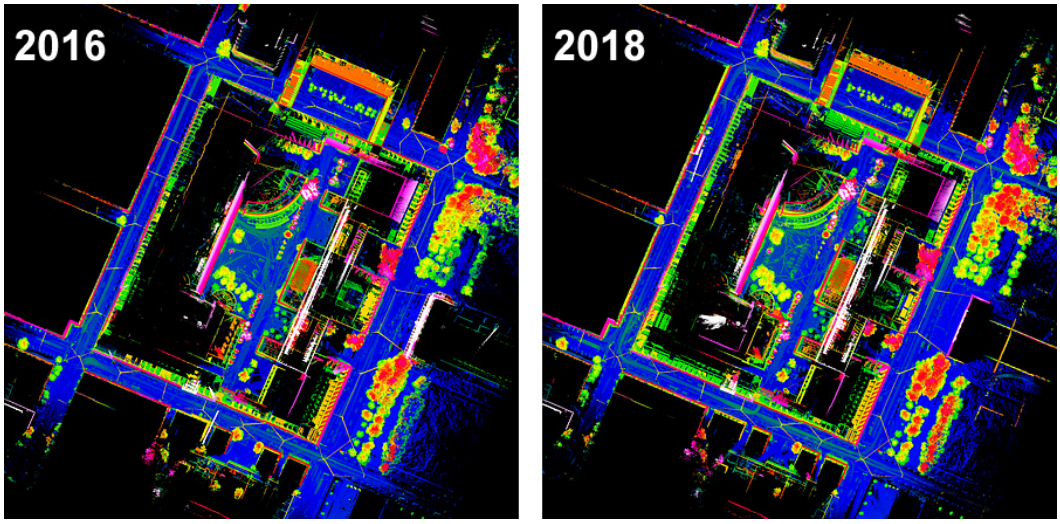
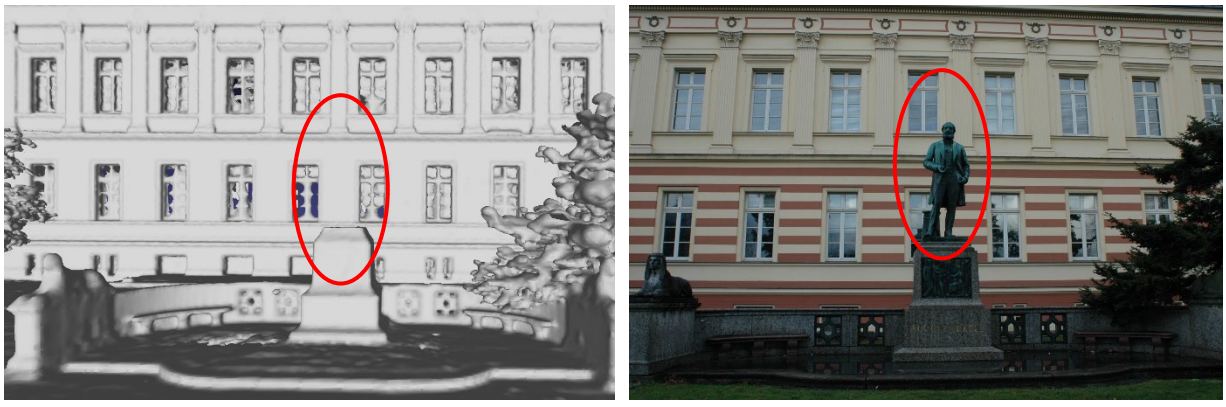


Figure 19. Top views of the dataset TUM-MLS-2016 and TUM-MLS-2018. Color code based on height. Source: Image adopted from [157].

URB3DCD. De Gélis et al. [158] proposed a bi-temporal simulated 3D dataset for CD in urban areas. They prepared five sub-datasets containing simulated pairs of 3D annotated point clouds with different characteristics, from high to low resolution, with various levels of noise [129].

The 2017 Change Detection Dataset. This is a dataset for visual CD consisting of images and 3D models. The dataset contains five different scenes. For each scene, it provides a 3D model and a set of images depicting a structural element not present in the model (see **Figure 20**). In addition, for each scene, it provides an XML file containing the calibration of the camera used to take the pictures, as well as the extrinsic poses in world coordinates. Finally, for each image, it also provides the ground truth obtained by manually labeling the areas of change [159].



(a) **(b)**

Figure 20. Finding changes in the environment based on an existing 3D model (a), and a sequence of (currently recorded) images (b).

We summarize in **Table 3** the different cited datasets we found in the state of the art. For each dataset we specify if it contains a classification (class label, such as building, tree, etc.) and if it contains a ground truth about the change (change label, such as

changed, unchanged, etc.). We also add, when available, the dates of acquisition and the references to the works that created these datasets.

Table 3. Summary of the existing datasets.

Dataset	Class Label	Change Label	Years	Reference
OpenTopography			Multiple years	[146], [160]
AHN1, AHN2, AHN3, AHN4	X		Multiple years	[149]
Abenberg—ALS test dataset	X		2008–2009	[150]
4D objects by changes	X		2017	[151], [152]
ICRA 2017—Change Detection Datasets			2017	[153]
PLS dataset of Kijkduin beach-dune			2016–2017	[154]
CG-PB-M3C2			2019	[161]
Near-continuous 3D time series				[155]
Change3D Benchmark		X	2016–2020	[128]
TUM City Campus—MLS test dataset	X		2009–2016–2018	[157]
URB3DCD	X	X		[129]
The 2017 Change Detection Dataset		X	2017	[159]

1.3.2 Evaluation metrics

The evaluation of the performance of a CD algorithm is a critical issue. CD algorithms process highly unbalanced data with respect to the ratio of changed to unchanged regions. Here, we introduce commonly used evaluation metrics. A well-known uncertainty metric for CD quantification is the level of detection (LODetection) [36], [162], [163], [164]. It is used as a threshold to consider only real changes (where the magnitude distance is superior to the LODetection at a specific confidence interval) for further analysis and interpretation (using a statistical t-test and an assumption of normal distribution of errors). It is determined by the number of points (n) in the point set and the variance of their distances to the fitted plane (σ^2), and by the registration error (alignment). It is defined by the following formula [139]:

$$\text{LODetection}_{\text{confidence interval}} = \pm 1.96 \times \left(\sqrt{\left(\frac{\sigma_A{}^2}{n_A}\right) + \left(\frac{\sigma_B{}^2}{n_B}\right)} + \text{alignement} \right) \quad (2)$$

where, A and B are the points clouds at epoch A and epoch B; σ^2 is the plane fitting variance (surface roughness); n is the number of points in the fitting neighborhood; alignment is the registration error, which can be estimated by using the absolute mean distance between two points clouds plus the standard deviation of these distance measurements.

Estimating the uncertainties through the process of CD is an important part of the workflow. It allows the differentiation between significant and non-significant changes. A change is considered significant when the quantified change magnitude is superior to

uncertainty (LODetection). A change is considered insignificant when the magnitude of change is inferior to, or equal to, the uncertainty.

To quantitatively evaluate change detection methods, there are several metrics used in the literature [119]. Two of these metrics, commonly used to compare the results of these methods with reference data, are correctness and completeness [119]:

$$\text{Correctness} = \frac{\text{TDN}}{\text{DN}} \times 100\%$$
$$\text{Completness} = \frac{\text{TDN}}{\text{RN}} \times 100\% \quad (3)$$

where TDN (True Detected Number) is the number of real changed objects (e.g., buildings) correctly detected as changed, DN (Detected Number) is the number of changed objects (e.g., buildings) in the detected results and RN (Reference Number) is the number of changed objects (e.g., buildings) in the reference data. Since these metrics take the object as a unit of comparison, as long as the detected object has an overlap with the reference object, it is considered as correctly detected [119].

The confusion matrix is another commonly used metric for quantitative analysis of binary or multiple classification. The common definition of the confusion matrix is presented in **Table 4**. Where FP (false positive) and FN (false negative) refer to the points that were incorrectly classified as changed and unchanged, respectively. TP (true positive) and TN (true negative) represent the changed points and unchanged points that were correctly detected, respectively.

Table 4. A simple example of confusion matrix of a binary change.

Detected	**Reference**	
	Changed	**Not Changed**
Changed	TP	FP
Not changed	FN	TN

From this matrix, we can derive the most used evaluation metrics in CD, which are overall accuracy, precision, recall, F1-score, and intersection over union, as shown in **Table 5**. Higher precision indicates fewer false prediction results, and higher recall indicates that fewer changes were missed. Furthermore, the larger their values are, the better the prediction results will be.

Table 5. Definitions of evaluation metrics for 3D point cloud CD.

Metric	**Description**	**Equation**
Overall accuracy	It is the general evaluation metric for prediction results.	$\text{OA} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}$
Precision	It measures the fraction of detections that were changed.	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$

Recall	It measures the fraction of correctly detected changes.	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$
F1 score	It refers to recall and precision together.	$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
Intersection over union	Or the Jaccard Index.	$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}$

The first type of metrics (LODetection) is used mainly for the standard approaches, while the confusion matrix and derived metrics in **Table 5** are used for learning approaches. Another important, but often neglected, metric is the calculation time, as well as the complexity of the workflow and the number of steps and operations it contains.

1.4 Discussion and Perspectives

In this review, we have subdivided 3D change detection methods into three main families. The first gathers traditional distance-based techniques that rely on direct geometric comparison. The second includes learning-based approaches that extract handcrafted features. The third comprises object-centric methods that integrate structural reasoning and temporal tracking. To provide a coherent synthesis, we proposed a structured taxonomy (see **Table 6**), organizing methods by processing level: pixel, point, voxel, segment, and object. Each category reflects a different level of abstraction, spatial support, and operational assumption regarding how change is perceived and modeled.

Pixel- and point-based methods remain the most direct, operating on raw or rasterized representations to measure elevation or displacement. They are well-suited for compact, dense data and straightforward geometric changes but are limited by their sensitivity to registration noise, surface roughness, and the absence of semantic context. Voxel-based approaches introduce regular spatial discretization, offering robustness to sparsity and facilitating volumetric comparison. Their grid-based logic, however, implies a trade-off between resolution and computational load. They are effective in large-scale, occlusion-prone environments where global consistency outweighs local precision. Segment-based methods shift the analysis toward mid-level structures, such as surfaces or clusters, providing a more stable unit of comparison. When segments are well-formed, they offer a meaningful abstraction that improves interpretability and reduces sensitivity to local variations. Yet, the quality of change detection remains tightly coupled to segmentation reliability, which can vary significantly depending on scene complexity. Object-based methods represent the most structured paradigm in the current taxonomy. They rely on explicit delineation, matching, and classification of real-world entities across epochs. This approach aligns naturally with applications requiring high-level interpretation, such as infrastructure monitoring or asset management. However, it also introduces dependencies on prior segmentation, parametric modeling, or semantic labeling, which may not be readily available or generalizable. To reveal the best-performing ones, in terms of metrics, a comparison is needed. de Gélis et al. (2021) performed a comparison between three methods: one based on distance calculation;

one based on machine learning with hand-crafted features and one based on deep learning [27]. Experimental results on the URB3DCD dataset showed that the machine learning method with hand-created features using random forest gave the best results, which, nevertheless, required supervision and a feature extraction step. Tao et al. (2021) conducted a benchmark of three methods, with one handcrafted feature (PoChaDeHH) and the other two learning-based (HGI-CD and SiamGCN) [128]. The results showed that the handcrafted algorithm had balanced performance over all classes. Learning-based methods achieved overwhelming performance but suffered from the class-imbalanced problem and might fail in minority classes. SiamGCN solved the class-imbalanced problem by adopting randomized oversampling and proposed a well-designed Siamese graph convolutional network architecture for the 3D CD. Comparison results showed that SiamGCN achieved the best performance on the released Change3D benchmark.

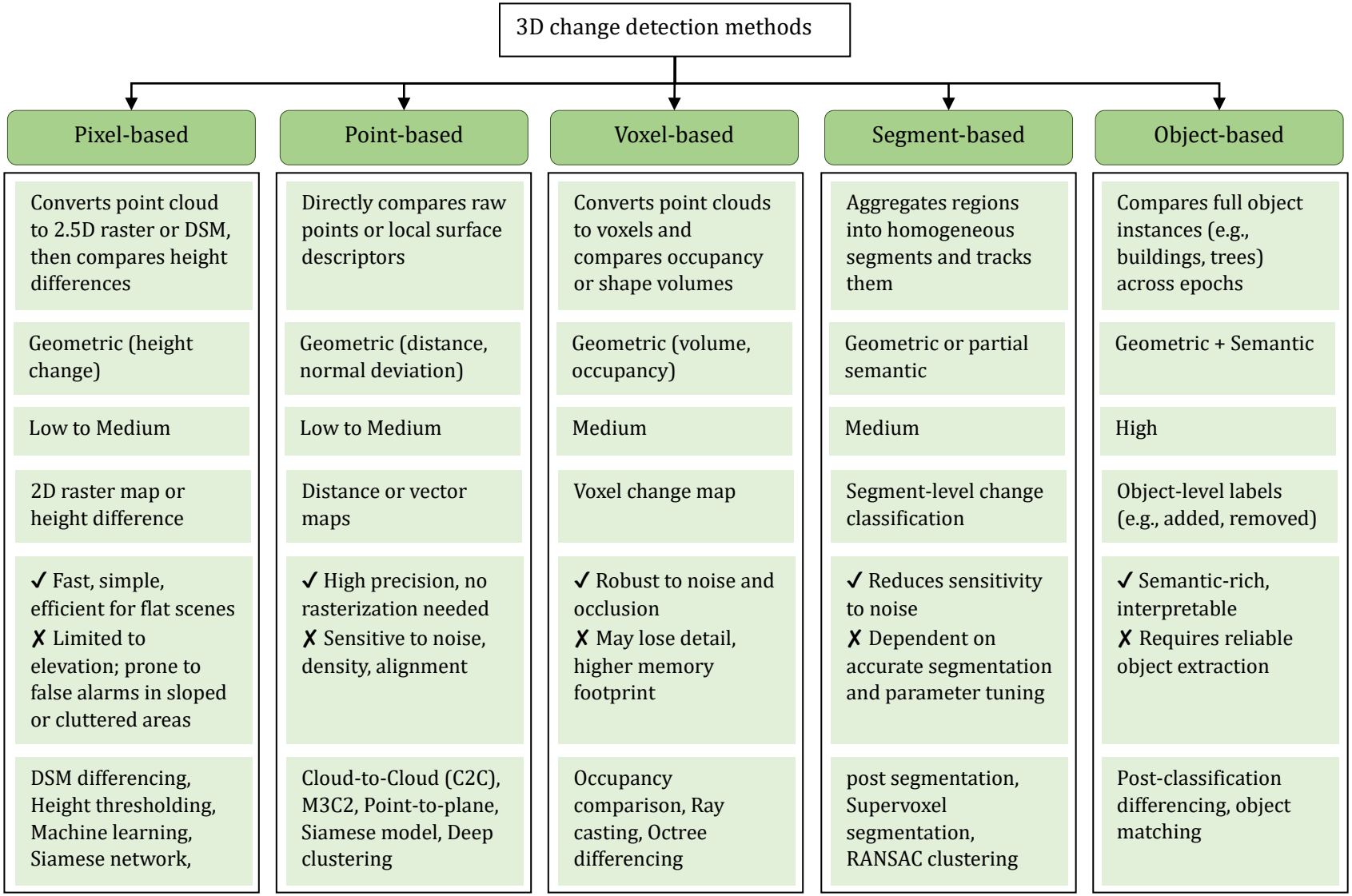
It is already well known that deep learning techniques show good performances in several point cloud processing tasks (semantic segmentation, object detection and recognition, object tracking, etc.). These techniques have demonstrated the same good performance over traditional methods in 3D CD tasks. Nevertheless, several problems are still related to this, which we summarize as follows:

Labeled data. Even though deep learning algorithms can learn highly abstract feature representations from raw 3D point clouds, successful detection and identification depend on large training samples. However, labeled high-resolution point cloud datasets are rare. Creating datasets for CD is more complicated than other tasks; it requires traversing point clouds not just at one epoch, but at two or more epochs. Given the lack of annotated data, to properly train the models, researchers use a variety of strategies, including transfer learning [165], [166], [167], data augmentation [166], [167], and Point Clouds Generative Adversarial Networks (PC-GAN) [168], [169]. Although these techniques alleviate some of the problems, coupled with a lack of samples, further improvements are still needed. Thus, future work should focus on creating methods that involve relying on small training datasets for supervised CD. This approach seems very interesting, as it minimizes the need for labeled training data.

Large-scale. In general, change happens in a reduced area and not on the entire point clouds. The lack of prior knowledge about the precise location of change, as well as the direction of change, means conventional unsupervised methods are unable to solve change quickly. More advanced studies are needed to solve this problem, such as hierarchical, weakly supervised, and semi-supervised methods.

Level of detail. Point clouds are massive data that can cover an entire country. These multi-temporal data collections are an ideal source for 3D change detection. The major problem is how to process such data, as change usually only occurs over limited areas. Level of detail structuring (LoD) is an efficient technique to address large data size. The idea is to use levels of detail from the highest to the lowest, so when more detail is required, the level of detail is lowered. The part that still needs to be studied is how the level of detail influences the results of the change detection. What metrics do we use to know the optimal LoD and to estimate the quality of the change accuracy?

Table 6. Comparative study using the main characteristics of 3D change detection approaches.



Thus, even if the challenges related to point cloud processing are overcome, due to the availability of processing techniques, there remain many other fundamental issues related to their use in change detection that need more investigation, mainly the following:

- The use of heterogeneous and multi-modal data (acquired by photogrammetry, laser scanner or other acquisition techniques).
- The use of multi-resolution data (acquired by sensors with different specifications).
- Handling near real-time laser scanning with a high temporal resolution that has become available today. Scene flow methods can play an important role in handling this data [56], [57], [58].
- The availability of benchmark data for the 3D CD.
- The exploitation of the progress made in the 3D semantic segmentation to integrate this information in the 3D CD process.
- The use of graph neural network for change detection [23], [170], [171].

Finally, it is worth pointing out that 3D CD is an active research field, which aims to reach robust methods to recognize dynamics and changes in any environment using dense point clouds. This literature review shows that this is a field that requires further research to improve the performance and accuracy of the CD results. The construction of these processes from point clouds data requires designing new approaches capable of detecting changes in the earth's surface with high accuracy and efficiency. Our future research tries to respond to these challenges by proposing, as a continuation of this work, a new approach that aims to enhance the quality of CD using prior semantics in 3D Point clouds. The objective is to improve the quality of true change and its characterization.

1.5 Conclusions

The detection of change and its characterization is an essential step for monitoring dynamics on the earth's surface. In this chapter, we presented a comprehensive review of change detection (CD) using 3D point clouds. We reviewed the methods used in the literature, and proposed several ways to classify them, and we highlighted the advantages and disadvantages of each category compared to the others. We proposed a first categorization based on classification and CD steps. Some start with CD and then classification, others do the reverse, and others integrate the two steps at the same time to avoid error propagation from one step to the next one. The second categorization is based on the algorithm used. A third proposed classification categorizes existing methods to ones based on distance (C2C, C2M, M3C2, CD-PB-M3C2), machine learning, and deep learning. We also revised the evaluation metrics for CD. Two categories are shown, statistical (e.g., LODetection) and classification metrics derived from the confusion matrix. The first one is for methods that are based on the calculation of distances, and the others are for the evaluation of change types classes.

On one hand, with the rise of learning-based approaches, particularly deep neural networks, significant improvements have been achieved in the accuracy and consistency of 3D change detection. Nonetheless, these methods are often dependent on large,

annotated datasets and suffer from reduced interpretability due to their complex architecture. Additionally, most existing methods operate at the point level, which, despite offering fine-grained detail, remains sensitive to noise and lacks structural coherence.

On the other hand, object-based strategies attempt to address these limitations by aggregating information over spatially consistent regions. However, their effectiveness is closely tied to the quality of segmentation and the discriminative ability of the extracted features. A critical limitation persists: the insufficient integration of semantic and geometric information within object-level change detection. In most existing works, semantic segmentation and change detection are performed as separate steps, or geometric change indicators are used without exploiting the semantics of the scene. This gap motivates the methodology proposed in the following chapter. We introduce an object-based framework that combines semantic segmentation and geometric descriptors in a unified change detection pipeline. The approach leverages spatial clustering to define coherent objects and integrates semantic labels with geometric indicators to characterize change types. This strategy improves robustness to noise and enhances the interpretability of results, while directly addressing the second research sub-question of this thesis: **How can semantic and geometric features be fused at the object level to improve change classification accuracy?**

CHAPTER 02:

Semantic and Geometric Fusion for 3D Change Detection

Preface

Building upon the foundations laid in **Chapter 1**, this chapter addresses one of the central research sub-questions of the thesis: ***How can semantic and geometric features be fused at the object level to improve change classification accuracy?***

While point-based and deep learning methods offer precise results, they often face difficulties in terms of spatial coherence, interpretability, and computational cost. In contrast, object-based change detection offers a more structured and efficient alternative, particularly in urban monitoring scenarios where changes often occur at the object level. This chapter presents a supervised object-based method that integrates semantic segmentation and geometric feature analysis to detect and classify changes in LiDAR point clouds.

The work described here was developed as part of a scientific publication (*Kharroubi et al., 2025*) and has since been expanded and contextualized to fit within the broader objectives of the thesis. It presents a complete pipeline involving semantic classification of bi-temporal point clouds, object clustering using the Cut-Pursuit algorithm, object matching across epochs, feature extraction, and change classification using a Random Forest model. Change types include both structural and natural changes (e.g., new buildings, demolitions, vegetation growth), making the approach particularly relevant for applications involving dynamic urban environments.

The method was evaluated on the Urb3DCD-v2 dataset, a simulated and publicly available benchmark for urban 3D change detection. The results demonstrate strong classification performance and computational efficiency. In particular, the integration of semantic labels and geometric descriptors (such as sphericity, omnivariance, and surface variation) improved interpretability and reduced classification noise compared to point-level methods.

The conclusions here directly support the domain-specific developments in the next chapters, including change-driven 3D model structuration and vegetation monitoring along railways. Together, they illustrate how semantic-geometric fusion can bridge the gap between low-level data and high-level understanding in 3D spatial analysis.

Based on the article (Kharroubi et al. 2025)

Semantic and Geometric Fusion for Object-Based 3D Change Detection in LiDAR Point Clouds

Kharroubi A, Remondino F, Ballouch Z, Hajji R, Billen R. *Remote Sensing. 2025.*

Abstract: Accurate three dimensional change detection is essential for monitoring dynamic environments such as urban areas, infrastructure, and natural landscapes. Point-based methods are sensitive to noise and lack spatial coherence, while object-based approaches rely on clustering, which can miss fine-scale changes. To address these limitations, we introduce an object-based change detection framework integrating semantic segmentation and geometric change indicators. The proposed method first classifies bi-temporal point clouds into ground, vegetation, buildings, and moving objects. A cut-pursuit clustering algorithm then segments the data into spatially coherent objects, which are matched across epochs using nearest-neighbor's search based on centroid distance. Changes are characterized by a combination of geometric features—including verticality, sphericity, omnivariance, and surface variation—and semantic information. These features are processed by a random forest classifier to assign change labels. The model is evaluated on the Urb3DCD-v2 dataset, with feature importance analysis to identify important features. Results show an 81.83% mean intersection over union. An additional ablation study without clustering reached 83.43% but was more noise-sensitive, leading to fragmented detections. The proposed method improves efficiency, interpretability, and spatial coherence of change classification, making it well-suited for large-scale monitoring applications.

Keywords: 3D change detection; point clouds; semantic segmentation; cut-pursuit; object-based; point-based; LiDAR

2.1 Introduction

Three dimensional change detection (3DCD) plays a crucial role in monitoring dynamic changes within a geographic area by analyzing multi-epoch data acquired using LiDAR (Light Detection and Ranging). In contrast to 2D remote sensing methods that are based on images, 3DCD uses 3D point clouds to capture structural and vertical modifications, such as new constructions, vegetation growth, and infrastructure degradation. This capability has broad applications in urban planning, environmental monitoring, infrastructure management, and disaster response [35], [172] .

Standard 3D change detection methods primarily classify changes into binary categories: changed and unchanged areas. The most common approaches, such as Cloud-to-Cloud (C2C) and Multi-scale Model-to-Model Cloud Comparison (M3C2), rely on spatial distance calculations to quantify differences between point clouds [139]. While effective in detecting geometric variations, these methods face several limitations, including sensitivity to sensor noise, variations in point density, and occlusions. More critically, they do not provide semantic information about the nature of detected changes, which is essential for real-world applications. To address these shortcomings, machine learning-based approaches have integrated feature extraction and classification into the change detection process. Tran et al. (2018) proposed a method to integrate classification and change detection in a single step using machine learning applied to airborne LiDAR [130] . It extracts features describing point distribution, terrain elevation, and multi-target capabilities, along with inter-epoch features that compare local point distributions between two epoch to detect changes. A supervised classifier then assigns each point to a change category. This approach avoids error propagation from sequential classification and change detection. However, it relies heavily on training data quality, struggles with subtle or gradual changes, and is sensitive to point density variations, and occlusions.

To improve robustness and interpretability, object-based change detection approaches have emerged, segmenting point clouds into meaningful entities before analyzing changes. Object-based methods enhance spatial coherence, reduce noise sensitivity, and allow for more detailed interpretation of change types. Zhang et al. (2023) proposed a framework that segments buildings using region growing and graph-cut techniques guided by point-level change indicators. It demonstrated good classification performance for complex urban changes on building [173]. Similarly, Stilla and Xu (2023) emphasized that aggregating points into coherent segments can reduce false detections, although segmentation accuracy remains a challenge, particularly in dense urban environments with varying point densities and occlusions [7]. Despite this progress, these approaches often lack integration with semantic segmentation methods and neglect the role of clustering quality in change detection performance. However, recent advancements in object-level semantic segmentation [6], such as superpoint-based methods, have proved the advantages of object-based processing. Robert et al. (2023) demonstrated that processing point clouds at the object level can improve computational efficiency, achieving up to a sevenfold increase in speed compared to existing methods [174]. Applying this principle to 3DCD suggests potential benefits for urban monitoring and rapid disaster response.

Concurrently, semantic change detection (SCD) [127], [175], [176] has advanced beyond binary classification by separating changes according to object categories (**Figure 21** shows the possible combination of semantic segmentation (SS) and change detection). Deep learning-based approaches, including Siamese PointNet [131], Siamese Graph convolutional network (SiamGCN) [177], End to end change detection [178], Siamese Kernel Point Convolution (Siamese KPConv) [179], Multi-task Enhanced Cross-temporal Point Transformer (ME-CPT) [180], and Prior-Knowledge-Guided Network for Urban 3D Point Cloud Change Detection (PGN3DCD) [133] have demonstrated strong performance by using hierarchical feature extraction and attention mechanisms. These methods excel at capturing fine-grained 3D changes but rely heavily on large-scale annotated datasets, which remain scarce. Thus, the lack of a comprehensive, large-scale benchmark dataset for 3D change detection forces researchers to rely on synthetic datasets, such as Urban 3D Change Detection version 2 (Urb3DCD-v2) [179], or manually annotated datasets, limiting model generalization. Additionally, a fundamental trade-off remains between the interpretability of traditional geometric methods and the data-driven nature of learning approaches. Object-based methods can present an alternative that balances interpretability, efficiency, and robustness, as they detect changes at the object level rather than at the individual point level. By using prior knowledge of object structures and spatial relationships, object-based approaches reduce dependency on large-scale annotated datasets and enhance generalization in real-world scenarios, particularly in urban settings where objects (e.g., buildings, vehicles) serve as primary units of interest.

Given these challenges, particularly the need for computational efficiency, interpretability, and robust semantic change identification, this chapter proposes a semantic-assisted object-based 3D change detection framework. Our approach integrates graph-based clustering with geometric and semantic feature fusion to provide robust, scalable, and interpretable change detection. It mitigates issues related to segmentation errors, computational complexity, and limited semantic interpretability prevalent in existing methods. To summarize, the main contributions of this work are to:

- Introduce a novel semantic-based object-based 3D change detection framework that integrates graph-based clustering with geometric and semantic feature fusion.
- Conduct a systematic ablation study to evaluate the influence of clustering on the overall performance of the change detection task.
- Provide a comprehensive benchmark with state-of-the-art methods.

The remainder of this chapter is organized as follows: Section 2 describes the proposed framework in detail. Section 3 presents experimental setup, and evaluation results, including an ablation study. Section 4 discusses the findings and limitations. Finally, Section 5 summarizes the chapter and outlines potential directions for future research.

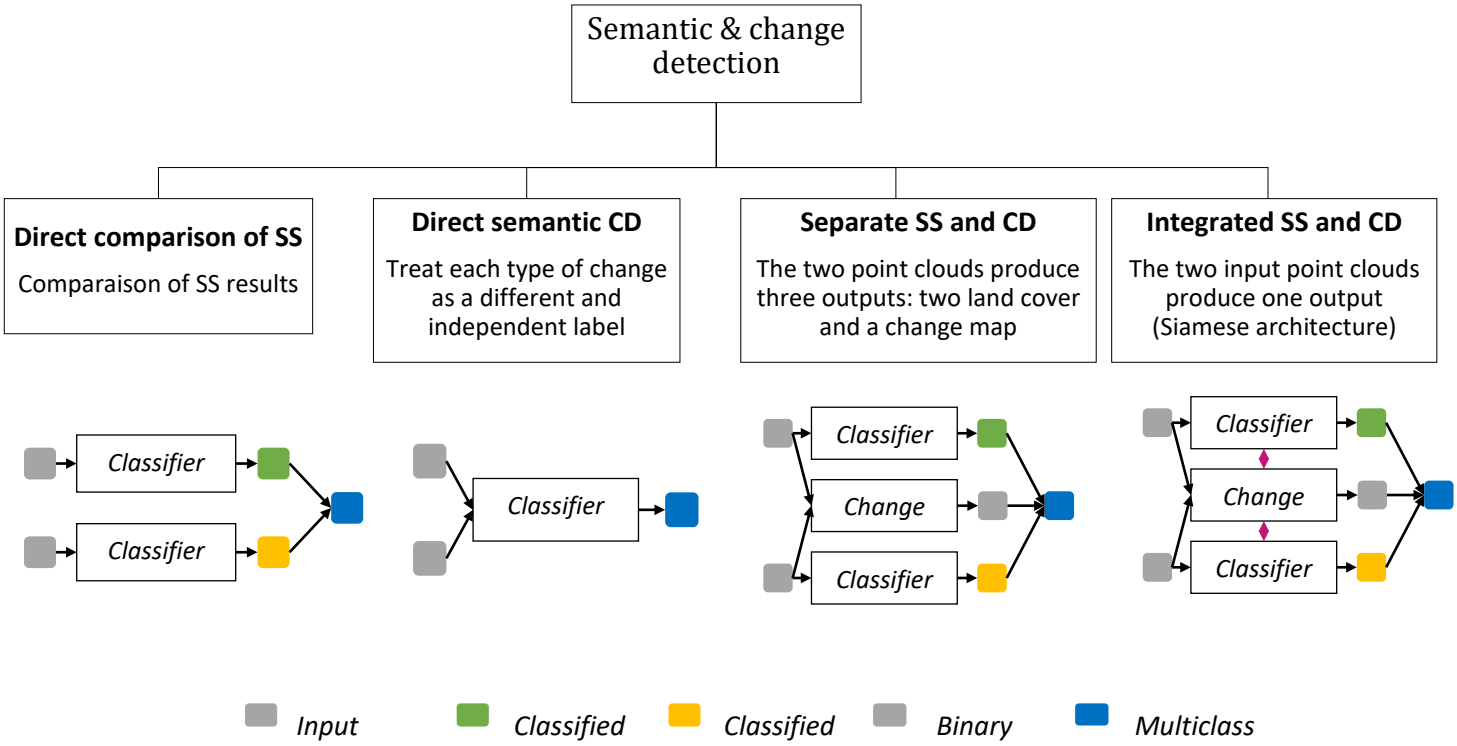


Figure 21. Illustrated categories of semantic change detection.

2.2 Materials and Methods

The proposed framework integrates semantic segmentation, geometric change indicators, and object-level analysis to improve the accuracy and consistency of 3D change detection. It follows a structured six-step process. First, preprocessing ensures comparability between bi-temporal point clouds through coordinate alignment, noise filtering, and spatial indexing. If the point clouds are unclassified, semantic segmentation assigns labels to ground, vegetation, buildings, and moving objects. Second, change indicators are extracted using displacement-based metrics to quantify geometric variations. Third, graph-based clustering segments the point cloud into spatially coherent objects, preserving geometric consistency. Fourth, object matching links corresponding objects across epochs using a nearest-neighbor search based on centroid distance. Fifth, feature extraction combines geometric descriptors (e.g., verticality, sphericity, omnivariance, and surface variation) with semantic information to characterize object-level changes. Finally, a random forest classifier assigns change labels, distinguishing different types of modifications. **Figure 22** summarizes the workflow of the proposed framework.

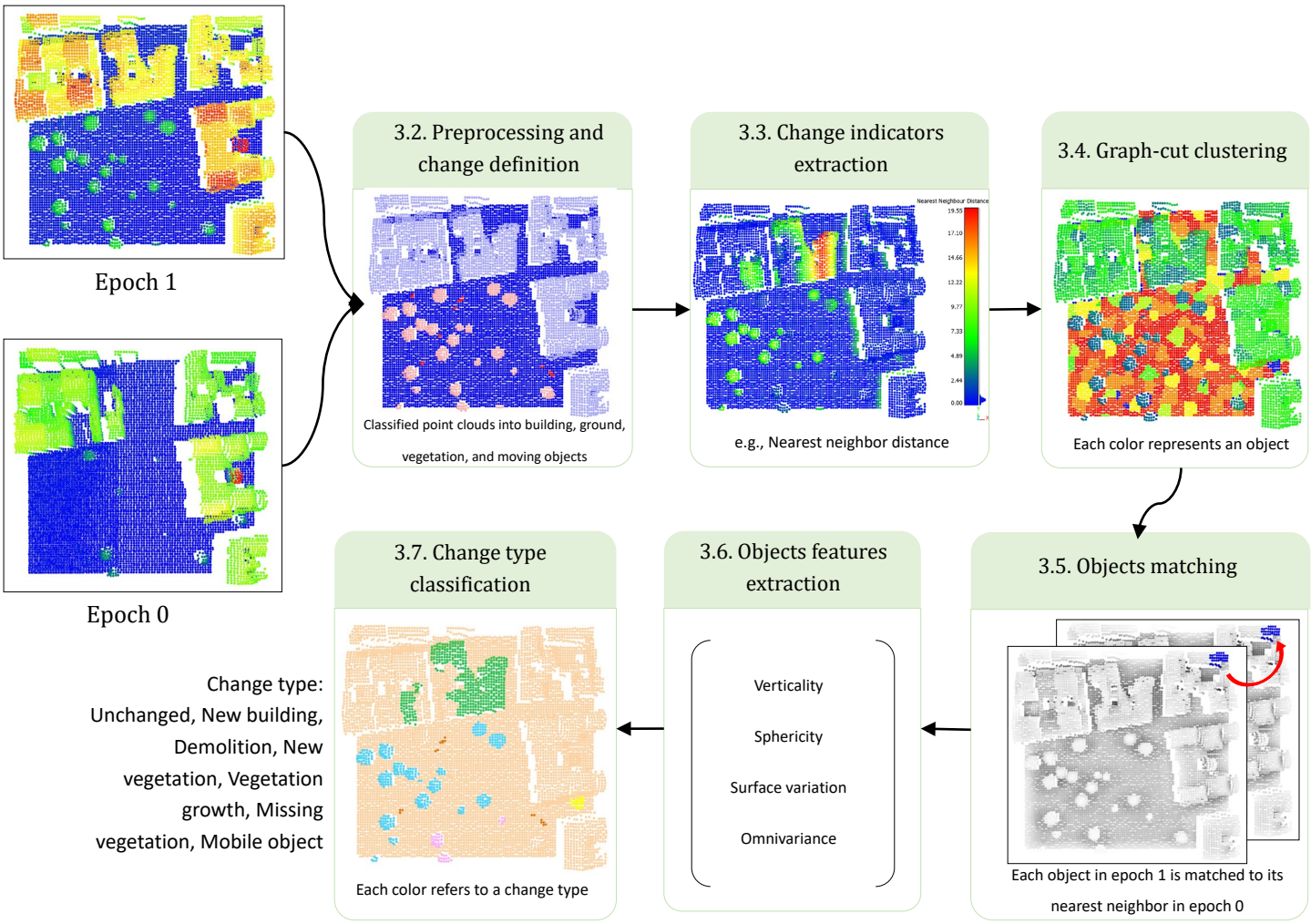
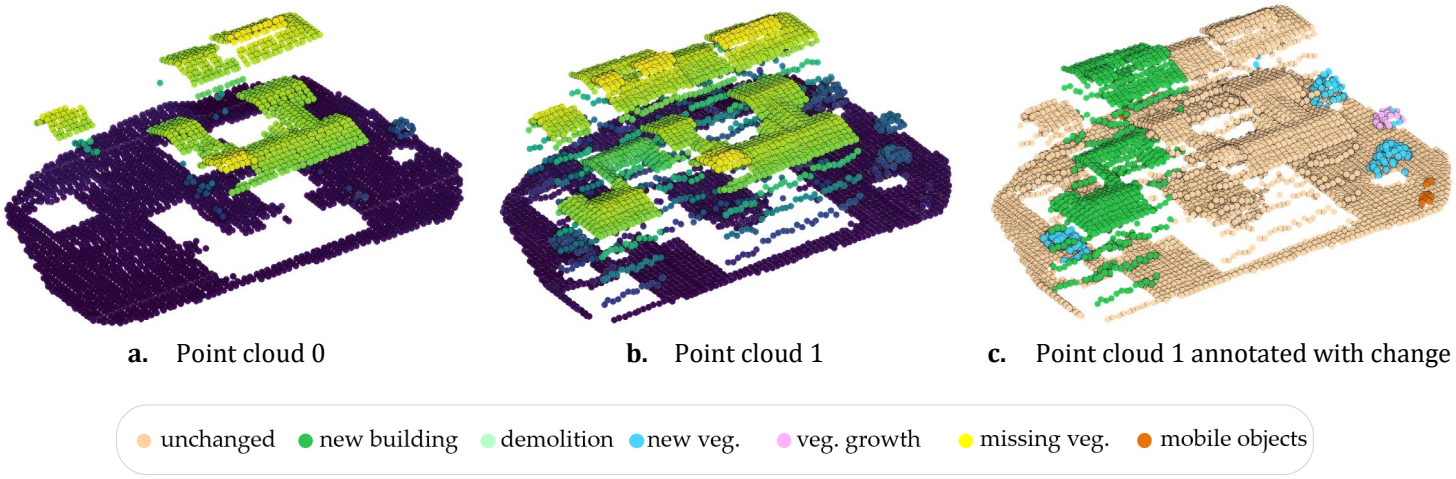


Figure 22. The illustrated methodology of our 3D change detection framework.

2.2.1 Simulated dataset

Urb3DCD-v2 is a publicly available simulated urban point cloud dataset designed for 3D change detection [179]. To the best of our knowledge, it is the only publicly available multi-class urban dataset that serves as a benchmark in the field [181]. It provides bi-temporal point clouds that simulate realistic urban change, ensuring a standardized evaluation for change detection methods. Each point is labelled into one of seven categories: unchanged, new building, demolition, new vegetation, vegetation growth, missing vegetation, and mobile objects. Additionally, semantic labels categorize points into ground, building, vegetation, and mobile objects. The dataset includes two subsets. The first subset has a point density of 0.5 points/m^2, simulating sparse urban acquisitions. While the second (multi-sensor) subset presents a first epoch with low density and high noise (mimicking photogrammetry), and a second epoch with higher density and lower noise, (resembling aerial LiDAR). **Figure 23** illustrates labeled ground truth examples, showing the distribution of change categories.



a. Point cloud 0 **b.** Point cloud 1 **c.** Point cloud 1 annotated with change

Figure 23. Clipped area from the simulated urban point cloud dataset: (a) Point cloud at epoch 0, colored by height; (b) Point cloud at epoch 1, colored by height; (c) Ground truth changes labels on point cloud 1.

It offers a controlled yet diverse evaluation environment, ensuring reproducibility while maintaining real-world complexity. The dataset is divided into training, validation, and testing sets, following a standardized protocol. **Table 7** summarizes key dataset characteristics, including point density, and class distributions.

Table 7. Summary of the dataset characteristics, including point density, and points distributions.

Dataset	Density	Number of tiles	Train points	Test points	Validation points
Urb3DCD-v2 (ALS Subset)	0.5 points/m^2	10 (Train), 3 (Test), 2 (Validation)	~ 3.2 million	~ 1.8 million	~ 1 million

A key feature of Urb3DCD-v2 is the simulation of occluded regions, a frequent challenge in urban LiDAR acquisitions. By varying the flight path of ALS (Airborne Laser Scanning) acquisitions, occlusion patterns shift between epochs, introducing realistic variations that complicate change detection. Occluded facades and ground areas reflect real-world sensor limitations. The dataset also captures long-term environmental changes, including tree growth, removal, and mobile object displacements. **Figure 24** presents the dataset's spatial distribution, showing tile placement and a zoom-in on a selected tile to illustrate spatial coverage and resolution. The classes used labels including ground, building, vegetation, and mobile objects.

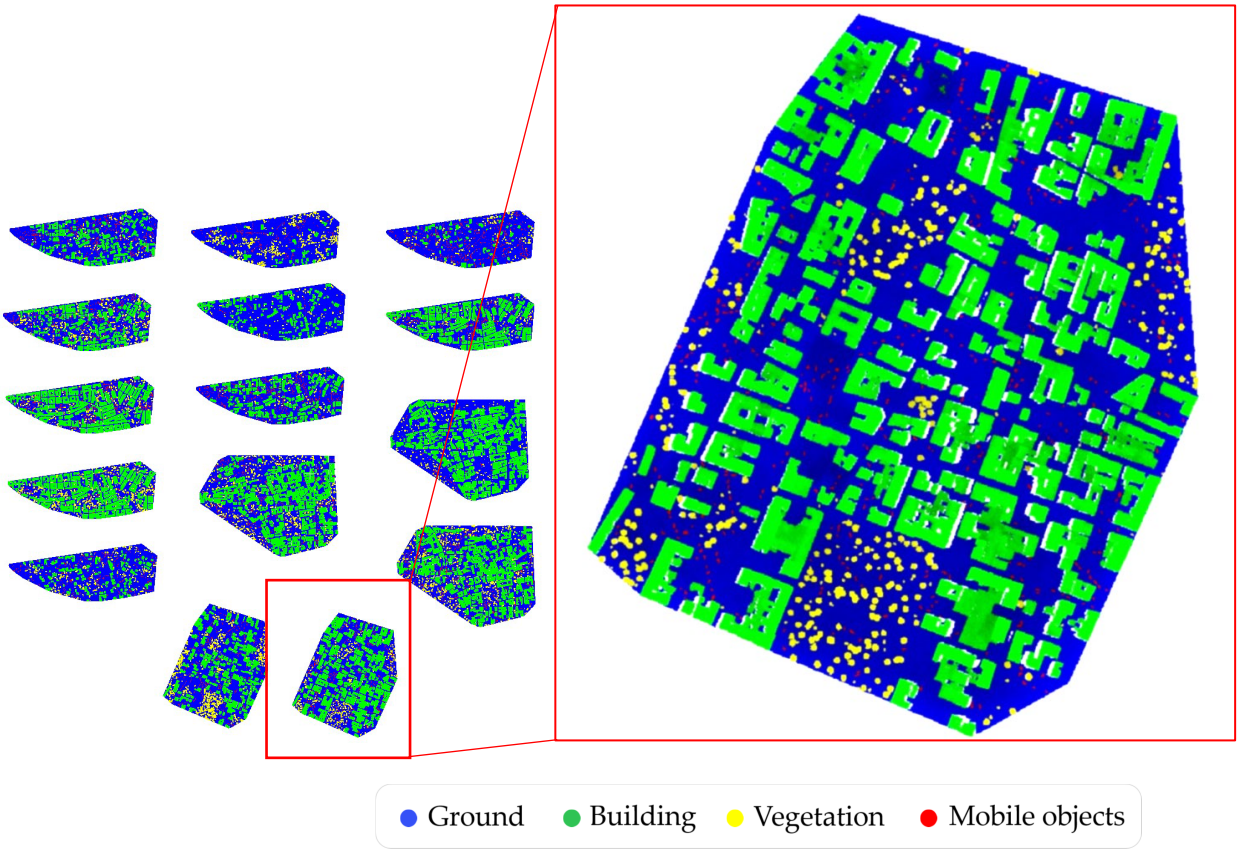


Figure 24. Overview of the dataset's spatial coverage with classes labels.

2.2.2 Preprocessing and change definition

Preprocessing ensures that bi-temporal point clouds are aligned and structured for reliable change detection. It generally includes coordinate alignment, noise filtering, and spatial indexing, as well as semantic classification to remove irrelevant variations. However, in this study, these steps were not performed, as the dataset already provides preprocessed data in a consistent coordinate system with predefined semantic labels. This classification helps refine change detection by filtering out irrelevant changes, such as temporary objects, which do not indicate meaningful change. Here, we categorize change relevance into three types: relevant, irrelevant, and falsified change. Relevant changes include modifications such as new or demolished buildings, height change, and tree growth or trimming [182], [183]. Irrelevant changes include temporary objects or classes in which we are not interested. Falsified changes, caused by sensor noise, misclassification, or minor misregistration, also introduce uncertainty into the process.

Formally, let Pt and Pt' two registered point clouds, acquired at times t and t' ($t \neq t'$), an object presents in Pt but absent in Pt' is classified as removed when $t < t'$, or new when $t > t'$. For unchanged or modified objects, our fusion framework of geometric and semantic features can determine the type of change based on the extracted features. The changes we are interested in are unchanged, new building, demolition, new vegetation, vegetation growth, missing vegetation, and mobile objects.

2.2.3 Change indicators extraction

Change indicators provide essential element for detecting modifications across epochs [184]. To capture displacement at multiple levels, we extract three key metrics: Cloud-to-Cloud (C2C) distance, planar C2C-2D, and Multiscale Model-to-Model Cloud Comparison (M3C2) distance. C2C distance measures direct point-wise displacement, computing the shortest Euclidean distance between corresponding points in both epochs. While computationally efficient, it is sensitive to variations in point density and occlusions, which can introduce false detections. To mitigate this, we introduce planar C2C-2D, which refines displacement estimation by projecting points onto a horizontal plane. This method ensures a more accurate assessment of lateral shifts in buildings and vegetation while filtering out vertical noise from LiDAR artifacts, such as façade points. However, both C2C and C2C-2D may lack robustness when dealing with non-rigid deformations (vegetation growth), prompting the integration of M3C2 distance. M3C2 estimates displacement along the local surface normal, fitting a cylindrical neighborhood around each point and computing the mean surface position per epoch. This approach provides a more stable measurement by accounting for surface roughness and registration uncertainties, reducing false positives in highly dynamic areas. These three complementary metrics form the geometric foundation of our change detection framework [139].

2.2.4 Graph-cut clustering

Our graph-cut clustering approach is based on the ℓ_0 cut-pursuit algorithm [185], which segments the point cloud into spatially coherent objects (a simplified principal shown in **Figure 25**). The point cloud is represented as a graph, where nodes correspond to individual points, and edges define connectivity based on spatial proximity and geometric similarity. Clustering is performed in two stages. First, local-scale segmentation groups points into fine-scale clusters using k-nearest neighbors (k-NN) and ℓ_0 cut-pursuit optimization. This ensures that each cluster preserves its geometric properties. Next, a global aggregation stage refines segmentation by merging clusters based on spatial consistency, preventing over-segmentation. The regularization parameter (λ) plays a key role in controlling the trade-off between segmentation granularity and spatial smoothness. A higher λ leads to fewer, larger clusters by enforcing spatial coherence and reducing unnecessary segment boundaries (under-segmentation). Conversely, a lower λ results in more granular segmentation, capturing small-scale variations but increasing segmentation complexity (over-segmentation). Optimization iteratively minimizes an energy function, to balance data fidelity and regularization constraints for an optimal segmentation. Additionally, the k-NN parameter influences spatial connectivity, to have structurally meaningful clusters. This adaptive segmentation method is particularly effective for detecting objects of varying scales [186], making it well-suited for this work. We maintain an over-segmented representation rather than merging all segments, to avoid change misclassification.

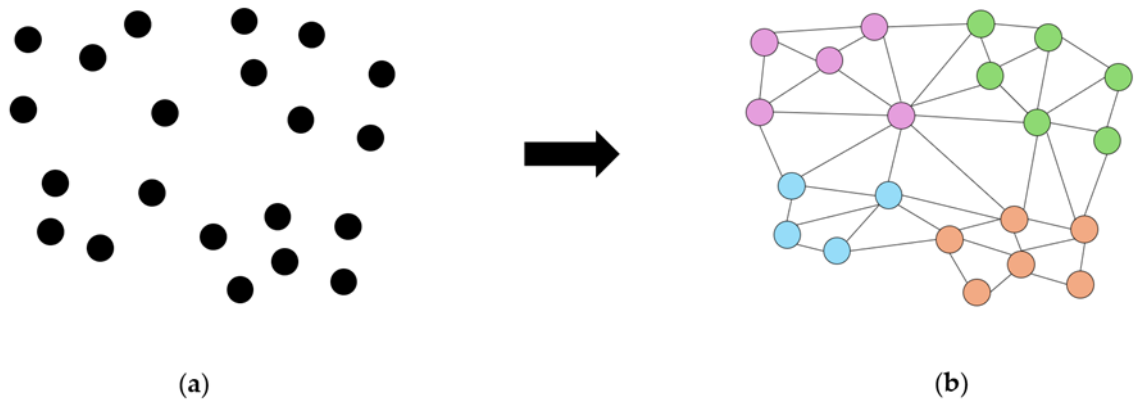


Figure 25. Illustration of partitioning a point cloud graph into clusters via ℓ_0 Cut-Pursuit [185]. (a) Initial point cloud; (b) Clustered point cloud graph with nodes, edges and distinct clusters represented by different colors.

A comparison with two widely used clustering methods, region growing and DBSCAN, is presented in the table below. The evaluation focuses on properties relevant to object-based processing, including spatial coherence, scalability, adaptability to multiple scales, and suitability for temporal change analysis. The ℓ_0 Cut Pursuit algorithm provides a favorable balance across all examined criteria. In contrast, region growing is constrained by its reliance on local normals and fixed-scale neighborhood definitions, which affect its robustness in heterogeneous scenes. DBSCAN shows better noise handling in uniform density settings but produces irregular cluster shapes and is sensitive to parameter selection. Recent studies, such as [186], have also demonstrated the effectiveness of ℓ_0 Cut Pursuit for segmenting complex 3D structures in natural environments, supporting its relevance for object-level change detection in this work.

Table 8. Comparison of clustering algorithms for 3D point cloud partitioning

Criteria	ℓ_0 Cut Pursuit (Landrieu & Obozinski, 2017)	Region Growing	DBSCAN (Ester et al., 1996)
Scalability to large datasets	High: graph-cut based with efficient optimization over decimated clouds	Medium: iterative and local, but slower on dense scenes	Medium: fast with spatial indexing, but sensitive to parameters
Noise robustness	High: energy minimization enforces smooth boundaries, reduces over-segmentation	Low: very sensitive to point density variations	Medium: density-based, but fails in varying density scenarios
Spatial coherence	High: clusters are spatially connected via graph topology	Medium: relies on normals, affected by noise	High: forms dense clusters, but may create irregular shapes
Multi-scale adaptability	High: λ parameter controls fine vs. coarse segmentation	Low: single scale, lacks adaptability	Low: ε and *MinPts* need tuning per scale
Integration with semantics	Easy: can incorporate semantic similarity in edge weights	Complex: not directly compatible	Difficult: purely geometric
Object boundary accuracy	High: boundary regularization avoids fragmented clusters	Medium: may leak across object edges	Low: irregular shapes, especially on borders

Suitability for change detection	Excellent: supports stable object tracking across epochs with centroid matching	Moderate: cluster IDs may not persist well	Poor: lacks temporal consistency
Computational efficiency	High after downsampling (e.g., to 1 m resolution)	Low on large data	Medium

2.2.5 Objects matching

Building on the segmented objects, the next step is to establish correspondences between epoch 0 and epoch 1. Since segmentation is performed using the cut-pursuit algorithm, each object is assigned a unique segment identifier (ID). To efficiently link objects between epoch 0 and epoch 1, we use a Nearest-Neighbor Search (NNS) based on centroid distance. The latter identifies the closest corresponding object in epoch 0 for each object in epoch 1. To accelerate correspondence search, we construct a k-dimensional (k-d) tree for each epoch using the centroids of objects. These k-d trees recursively partition the 3D space along alternating coordinate axes (x, y, z), creating an efficient hierarchical structure for fast nearest-neighbor retrieval. Each object in epoch 0 is then matched to its nearest neighbor in epoch 1. Instead of using a fixed threshold to determine whether an object has disappeared or appeared, we treat centroid displacement as a feature to be used in the classification stage. The Euclidean distance between matched centroids serves as a change indicator, where larger distances suggest object displacement or modification. We used this adaptive approach to avoid rigid thresholds, allowing for more flexible and accurate change classification.

2.2.6 Object features extraction

Once objects are matched across epochs 0 and 1, feature extraction is performed to characterize their geometric and semantic properties, providing structured inputs for change classification. The extracted features capture both the shape and structure of objects, as well as their semantic consistency over time. Geometric features include verticality, surface variation, omnivariance, and sphericity [187]. Semantic features assess the stability of an object's classification across epochs, ensuring that changes are meaningfully detected. In addition to centroid displacement, these features are aggregated to compute the nearest feature difference, which serves as the final input to the classification model.

Aggregation of point-level and object-based features

Each object consists of multiple points, requiring feature aggregation to obtain meaningful object-level descriptors. For continuous features, such as verticality, surface variation, omnivariance, and sphericity, the mean value is computed per object. For example, mean verticality captures the overall orientation of an object, while mean surface variation reflects its roughness. For categorical features, such as the semantic label from epoch 1 and the propagated label from epoch 0, the median is used to ensure a robust representation of the dominant class within the object. This prevents outliers or mislabeled points from affecting the final classification. The following **Table 9** summarizes the features used in this study:

Table 9. Summary of aggregated point level and object-based features.

Feature Category	Feature Name	Description
Change indicators	Nearest neighbor distance (c2c)	Measures the shortest Euclidean distance to the nearest point in the other epoch, indicating positional change.
	2D nearest neighbor distance (2D c2c)	Computes nearest neighbor search constrained to the horizontal plane, reducing vertical noise.
	M3C2	Measures displacement along the local surface normal, providing a more reliable comparison for non-rigid changes.
	Distance uncertainty	Quantifies the confidence level of M3C2 displacement measurements, accounting for roughness and registration errors.
Geometric	Verticality (6m)	Evaluates the degree to which an object aligns with the vertical axis.
	Sphericity (6m)	Indicates how closely an object approximates a sphere.
	Surface Variation (6m)	Measures local roughness by analyzing normal vector dispersion.
	Omnivariance (6m)	Assesses the spread of points in all directions.
	Object center displacement	Computes the Euclidean distance between an object's centroid across epochs
Semantic	Class Label (Epoch 1)	Identifies the most frequent semantic class in the recent epoch.
	Class Label Propagation (Epoch 0)	Tracks how labels from the previous epoch propagate to the actual one.

Nearest feature difference

The suggested Nearest Feature Difference operation (NFD) quantifies the difference of object features between two epochs. It is computed for each object by comparing its feature values at epoch 1 to those of its nearest corresponding object in the other epoch. This metric captures geometric and semantic changes while preserving the directional nature of differences (which can be negative or positive). Let:

$\mathcal{O}_t = \{\mathcal{o}_1, \mathcal{o}_2, \dots, \mathcal{o}_n\}$ be the set of objects in epoch t. (4)

$\mathcal{O}_{t'} = \{\mathcal{o'}_1, \mathcal{o'}_2, \dots, \mathcal{o'}_m\}$ be the set of objects in epoch t'. (5)

$\mathcal{F}_t(\mathcal{o}_i) = (f_1, f_2, \dots, f_k)$ be the feature vector of object $\mathcal{O}_i$ at epoch t. (6)

$\mathcal{F}_{t'}(\mathcal{o'}_j) = (f'_1, f'_2, \dots, f'_k)$ be the feature vector of object $\mathcal{O}_j$ at epoch t'. (7)

$\mathcal{N}(\mathcal{o}_i)$ be the nearest object to $\mathcal{o}_i$ in epoch t', determined using a nearest-neighbor search. (8)

We define the NFD vector for $\mathcal{o}_\mathrm{i}$ as:

$$NFD\ (\sigma_i) = (f_1(\sigma_i) - f_1(\mathcal{N}(\sigma_i)), f_2(\sigma_i) - f_2(\mathcal{N}(\sigma_i)), \dots, f_k(\sigma_i) - f_k(\mathcal{N}(\sigma_i))) \quad (9)$$

Where:

- $f_p\ (\sigma_i)$: represents the pth feature of the object at epoch t.
- $f_p\ (\mathcal{N}(\sigma_i))$: represents the same feature for the nearest-matched object at epoch t'.
- k : is the total number of features.

Each NFD vector component preserves the sign of the difference, to allow the detection of both magnitude and direction of change.

2.2.7 Change type classification

We use a Random Forest (RF) classifier to classify object-level changes based on geometric and semantic features. RF is an ensemble learning method that combines multiple decision trees, where majority voting improves stability and enhances generalization. Unlike single decision trees, which are prone to overfitting, RF leverages bootstrap aggregation, training each tree on randomly sampled subsets of the data. This process increases robustness while reducing sensitivity to noise and outliers. Additionally, RF efficiently manages heterogeneous features. Its low computational cost and scalability further make it an attractive choice for large-scale 3D change detection. Several parameters influence RF's performance. The number of trees controls the trade-off between variance reduction and computational cost. The maximum depth regulates model complexity, preventing overfitting. Meanwhile, the split criterion, such as Gini impurity, determines the best feature at each node by maximizing class separability. This ensures that the classifier prioritizes the most discriminative features when distinguishing objects across epochs. RF also provides feature importance analysis, ranking attributes based on their contribution to impurity reduction across decision splits. This enables us to quantify the role of NFD in identifying new, disappeared, modified, and unchanged objects.

2.3 Results

To evaluate the proposed framework, we conducted experiments on the Urb3DCD-v2 dataset, assessing both quantitative and qualitative performance. The evaluation includes a comparison with state-of-the-art methods, an ablation study to measure the impact of object-based clustering, and a feature importance analysis to identify key contributors to change classification.

2.3.1 Implementation

All pre-processing, feature extraction, and classification tasks were conducted using Python, using key libraries such as Scikit-Learn for machine learning and the LiDAR Platform[1] for point cloud processing and automation. The cut-pursuit clustering was

[1] https://lidar-platform.readthedocs.io/en/latest/

implemented using the TreeIso library[2] [186], ensuring efficient clustering of large-scale point clouds. Additionally, custom scripts were developed to integrate data processing, object matching, and feature computation into a streamlined workflow. To accelerate object matching, we used the KDTree implementation from scipy.spatial, which efficiently structures centroid locations for nearest-neighbor searches. The experiments were executed on a high-performance workstation equipped with an NVIDIA GeForce RTX 3090 GPU, an Intel i9-10980XE CPU (3.00 GHz), and 256 GB of RAM (Random-Access Memory).

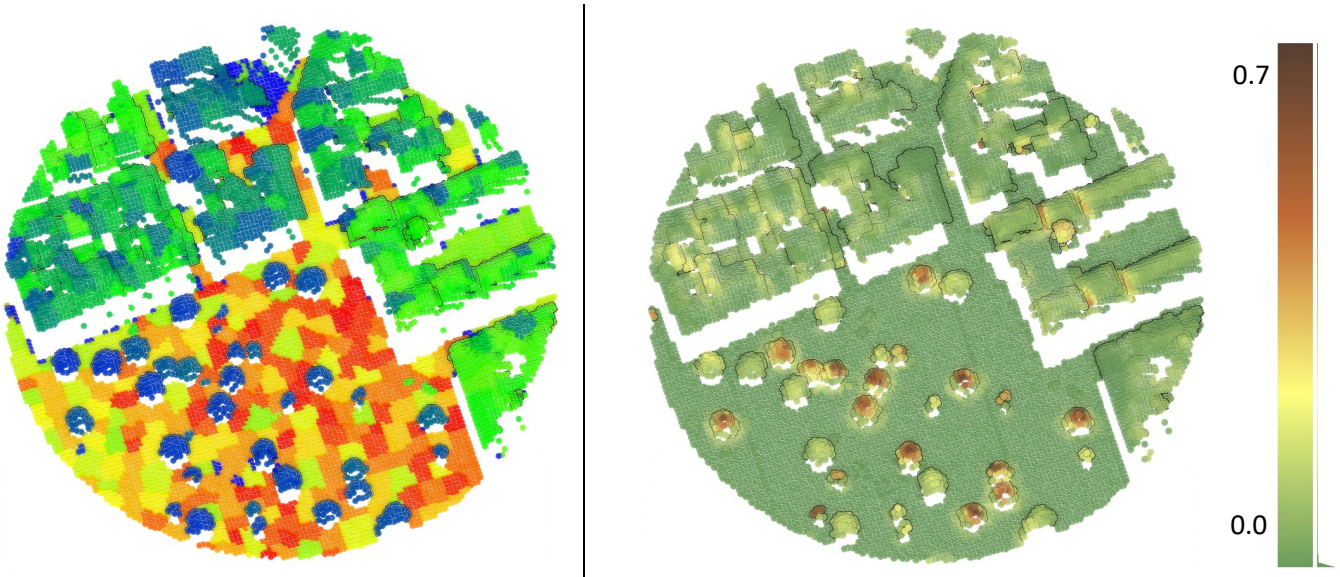
2.3.2 Experiment setting

The proposed method was implemented using the predefined point clouds classification, with all the data already registered. Feature extraction was performed using a 6-meter diameter for computing geometric features such as verticality, sphericity, surface variation, and omnivariance, to ensure robust local shape characterization. Change indicators were derived using C2C distances computed with nearest-neighbor search, and M3C2 distances with a normal diameter of 14 meters, a projection diameter of 14 meters, and a maximum depth of 50 meters to capture variations at multiple scales, particularly for complex urban structures and vegetation (see **Figure 26b**).

For clustering, the ℓ_0 cut-pursuit algorithm was applied with a regularization parameter $\lambda = 12$, balancing spatial consistency and cluster separation, while a 6-neighbor connectivity ensured that local geometric structures were preserved (see **Figure 26a**). To optimize computational efficiency, the point cloud was decimated to a 1-meter resolution, reducing processing time while maintaining sufficient detail for change classification. This choice was guided by prior research on 3D graph-based clustering (e.g., TreeIso method, Xi & Hopkinson, 2022) and validated through visual assessment to ensure robust object segmentation.

NFD was computed using nearest-neighbor search to quantify differences between corresponding objects across epochs. The features used in classification are detailed in **Table 9**. Random Forest classifier was trained with 150 trees, a maximum depth of 20, and a minimum of 10 samples per split, to have a balance between model complexity and generalization. These hyperparameters were determined using grid search cross-validation, optimizing for high classification performance while preventing overfitting.

[2] https://github.com/truebelief/artemis_treeiso/tree/main/Python



(a). Clusters with different colors

(b). Sphericity feature

Figure 26. Illustration of Cut-Pursuit clustering and sphericity feature results. (a) shows results of cut-pursuit clustering, where each color represents a distinct object. (b) shows an example of a geometric feature, sphericity, calculated at a 6-meter diameter. The color scale ranges from dark brown (high sphericity) to green (low sphericity), indicating how well the neighbors fit a spherical shape.

2.3.3 Evaluation metrics

The performance of the proposed method is evaluated using three standard classification metrics: recall, mean Intersection over Union (mIoU), and per-class Intersection over Union (IoU). These metrics provide an objective assessment of classification performance at both global and class-specific levels. Recall quantifies the proportion of correctly detected changes among all actual changes. Per-class IoU is computed for each change type, providing a detailed assessment of class-specific performance. Additionally, $mIoU$ measures the overall classification performance by computing the average overlap between predicted and reference change regions across all classes. Higher values of these metrics indicate better classification results. The equations for these metrics are as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{IoU}_\text{i} = \frac{\text{TP}_\text{i}}{\text{TP}_\text{i} + \text{FN}_\text{i} + \text{FP}_\text{i}} \tag{11}$$

$$\text{mIoU} = \frac{1}{\text{C}} \sum_{\text{i}=1}^{\text{C}} \text{IoU}_\text{i} \tag{12}$$

where C is the total number of classes, TP represents True Positives, FP False Positives, and FN False Negatives. These metrics collectively ensure a robust evaluation of change detection accuracy, both at the global and per-class levels.

2.3.4 Quantitative results

The **Table 10** presents the quantitative evaluation of our proposed object-based change detection framework across multiple change categories. Our method achieves an mIoU of 81.83% and a recall of 86.39%, demonstrating strong classification performance. Notably, it performs well in detecting major structural changes such as new buildings (87.93%) and demolition (81.40%), benefiting from the integration of geometric and semantic features. For vegetation-related changes, our method achieves 94.07% IoU for new vegetation, indicating its ability to distinguish between persistent and newly introduced natural elements. Our method outperformed the random forest method by [130] which achieve an mIoU of only 53.37% (see **Figure 31** for visual comparison). When compared to deep learning models such as Siamese KPConv and Triplet KPConv, our method remains competitive despite not relying on learned feature representations. While deep models achieve higher mIoU (84.08%), they require substantial computational resources. By using the handcrafted geometric and semantic features in our machine learning framework, we provide a scalable and interpretable alternative for 3D change detection. A detailed confusion matrix is presented in Appendix A, to offer a per-class percentage of classified objects.

We analyzed two 2D methods: DSM-Siamese (Digital Surface Model Siamese) and DSM-FC-EF (Digital Surface Model Fully Connected Network with Early Fusion) results. DSM-Siamese and DSM-FC-EF operate on rasterized digital surface models (DSM), detecting changes based on height variations between epochs. These methods perform well in structured environments but struggle with complex vegetative and mobile object changes due to their reliance on height differences rather than object-level features. In our evaluation, DSM-Siamese achieved an mIoU of 57.41%, while DSM-FC-EF reached 56.98%, showing limitations in handling diverse change categories, particularly vegetation growth (8.92% IoU for DSM-Siamese and 1.89% for DSM-FC-EF).

Additionally, we analyzed the DC3DCD EFSKPConv model [132] which introduces an unsupervised learning approach for multiclass 3D change detection, leveraging deep clustering principles to segment changes without requiring extensive labeled datasets. It extracts pseudo-clusters from raw point clouds, which are later mapped to real-world change classes through a weakly supervised learning process. Compared to traditional supervised models, DC3DCD reduces the dependency on annotated data, making it scalable for large-scale change detection tasks. The evaluation on Urb3DCD-v2 shows that DC3DCD achieves an mIoU of 57.06%, surpassing classical machine learning models such as Random Forest and DSM-based approaches. However, its performance remains lower than fully supervised deep networks, including Triplet KPConv (84.08%) and Siamese KPConv (80.12%). Implicit neural representation (INR) methods take a continuous function-based approach to change detection, encoding spatiotemporal variations through learned representations. These include Sinusoidal Representation Network (SIREN) and Random Fourier Features (RFF), both combined with Spatial Encoding (S), Total Variation Norm (TVN) for spatial smoothness, and Temporal Difference (TD) for temporal coherence. By reconstructing the underlying surface geometry at each epoch, these models aim to improve structural consistency over time. The experiments show that RFF+S+TVN+TD achieved a low per-class IoU,

underperforming DC3DCD but still falling short of fully supervised deep learning models. INR-based methods excel in capturing large-scale height variations, but their reliance on function approximation makes them less effective in detecting fine-scale object changes, such as small vegetation growth or mobile objects.

Table 10. Quantitative evaluation of methods on the simulated point cloud dataset.

			Per class IoU (%)						
Method	**Recall (%)**	**mIoU (%)**	*Unchanged*	*New building*	*Demolition*	*New veg.*	*Veg. growth*	*Missing veg.*	*Mobile object*
RF [130]	/	52.37	92.72	73.16	64.60	75.17	19.78	7.78	73.71
DSM-Siamese [125]	/	57.41	93.21	86.14	69.85	70.69	8.92	60.71	8.14
DSM-FC-EF [125]	/	56.98	94.39	91.23	71.15	68.56	1.89	62.34	46.70
Siamese KPConv [179]	85.31	80.12	95.82	86.67	78.66	93.16	65.18	65.46	91.55
Triplet KPConv [184]	86.41	**84.08**	**97.41**	**95.73**	81.71	**96.24**	**64.85**	73.02	92.90
DC3DCD EFSKPConv [132]	/	57.06	93.96	79.26	67.88	75.34	19.48	20.29	80.10
SIREN+S+TVN+TD [188]	/	/	84.83	62.62	47.92	4.26	0.62	3.89	0.26
RFF+S+TVN+TD [188]	/	/	87.47	71.81	57.63	5.44	1.54	8.82	0.58
Object-based (ours)	86.39	81.83	96.57	87.93	81.40	**94.07**	**57.01**	72.04	83.83
Point-based (ablation)	**86.99**	83.43	96.98	91.54	**82.31**	93.98	45.37	**73.86**	**100**

Compared to these approaches (see **Figure 31** for visual comparison), our object-based change detection framework provides a balanced trade-off between efficiency, interpretability, and robustness. Achieving an mIoU of 81.83%, our method outperforms unsupervised approaches while maintaining lower computational costs than deep learning models. It reduces false positives, particularly in new building detection (87.93% IoU), demolition (81.40% IoU), and new vegetation (94.07% IoU). While INR methods offer a novel perspective on spatiotemporal modeling, their computational complexity and sensitivity to function fitting limit their real-world applicability. Despite its strengths, our method can be further improved by integrating feature-learning techniques from deep clustering methods like DC3DCD while retaining the structural coherence of object-based segmentation.

A key advantage of our method is its computational efficiency. By operating at the object level rather than the point level, the method significantly reduces the number of elements processed, leading to faster training and inference times (it was 6 times faster than point based methods). In terms of interpretability, in **Figure 27** we present the feature importance analysis which highlights the key features contributing to change type classification. The x-axis represents the different features analyzed, while the y-axis represents the importance of each feature. The units on the vertical axis are dimensionless and represent the relative importance of each feature, normalized to a range between 0 and 1, where 1 indicates the most important feature. This normalization allows for a clear comparison of the contribution of each feature to the classification performance. The 2D nearest neighbor distance holds the highest importance (0.31), indicating that spatial proximity within the horizontal plane is a crucial factor for distinguishing changes. The semantic labels (label 0 and label 1) also play a significant role (0.12 and 0.14, respectively), emphasizing the relevance of object

class consistency across epochs. The M3C2 distance (0.06) and objects centroid displacement (0.06) provide valuable insights into geometric changes, particularly for structural modifications. Distance uncertainty (0.04), omnivariance (0.04), surface variation (0.04), and sphericity (0.04) contribute moderately while verticality (0.03) holds the lowest importance, suggesting that vertical alignment is less discriminative for identifying changes in this dataset.

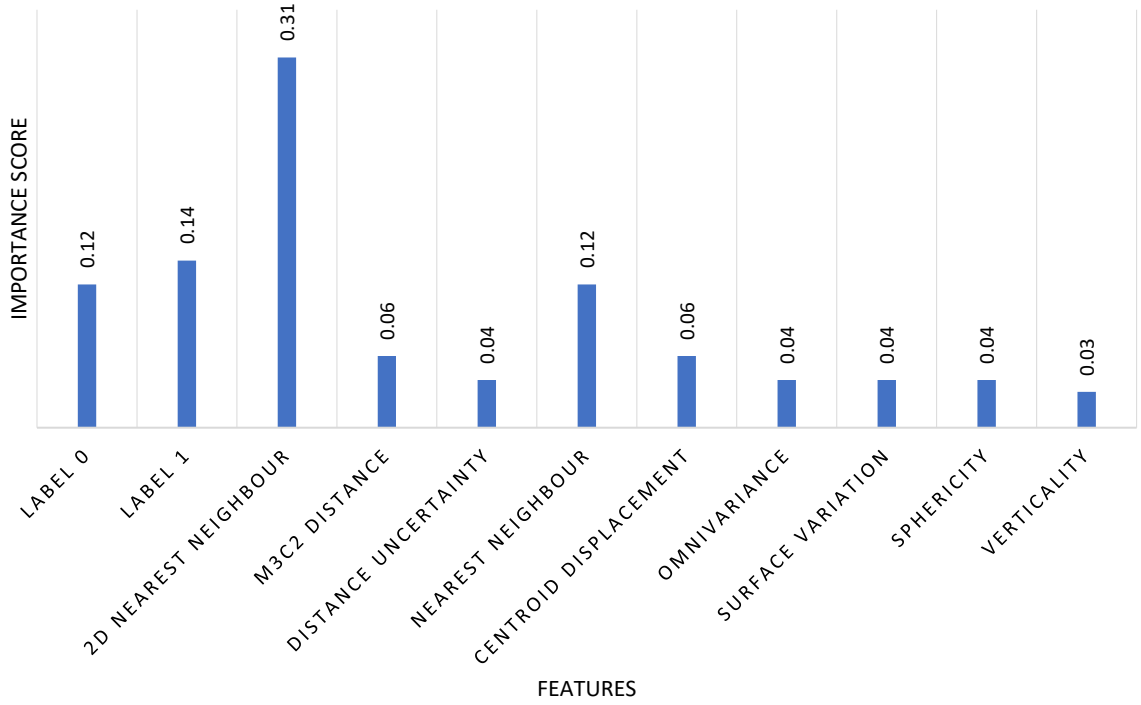


Figure 27. Contribution of geometric and semantic features to object-based change classification. The values represent the relative importance of each feature, normalized to a range between 0 and 1, where 1 indicates the most important feature.

2.3.5 Ablation study

To further experiment our work, we analyze the impact of clustering on change detection by evaluating change classification without object-based features or spatial constraints. This study isolates the effect of clustering on classification performance. The point-based method classifies points independently, without object-level clustering, leading to a lack of spatial coherence. The objective is to assess the impact of working at the point level, without segmenting objects beforehand or applying post-classification refinement. We used the same features and the same random forest classifier parameters. The results (in **Table 10**) show that the point-based classification achieves a mean Intersection over Union (mIoU) of 83.43%, with per-class IoU values ranging from 45.37% (vegetation growth) to approx. 100% (moving objects). The confusion matrix visualization (Annex) highlights that certain change classes, particularly class 4 (vegetation growth), suffer from reduced performance, likely due to local inconsistencies in point-based predictions. This suggests that spatially enforcing coherence, through object-level classification, improves robustness. Analysis of the feature importance indicates that 2D nearest neighbor search (28.91%) and label propagation from the first epoch (12.84%) are the most influential features, followed by

M3C2 distance (7.89%) and sphericity (6.08%) see **Figure 28**. This suggests that geometric change features, particularly those using bi-temporal comparisons, are important for changes.

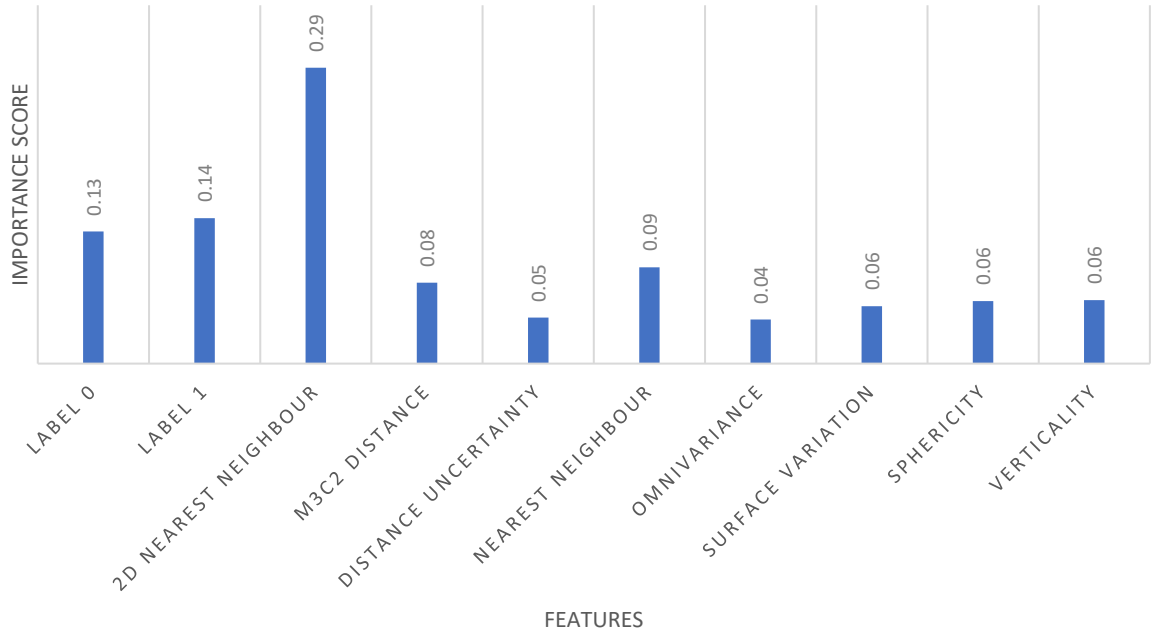
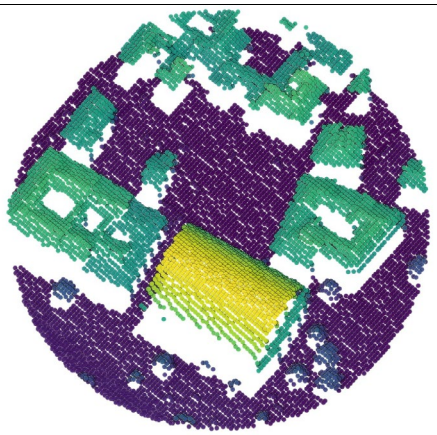
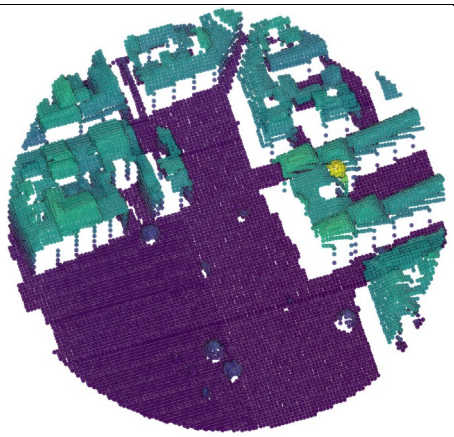
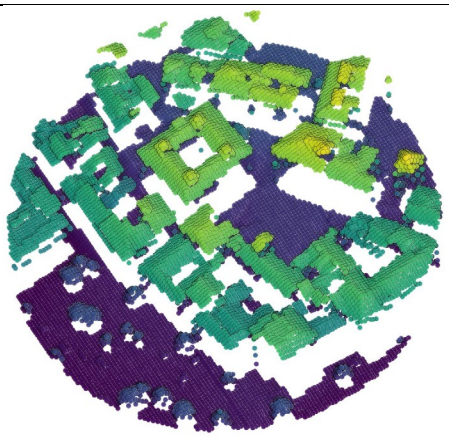


Figure 28. Contribution of geometric and semantic features to point-based change classification. The values represent the relative importance of each feature, normalized to a range between 0 and 1, where 1 indicates the most important feature.

2.3.6 Qualitative results

The qualitative evaluation compares the predictions of the object-based and point-based methods with the ground truth (**Figure 29**). The point-based approach captures detailed changes with high precision but introduces noise (red circles), as small variations are sometimes misclassified as change. A main limitation appears in vegetation growth detection, where part of the tree is classified as unchanged, while new points are labelled as growth. This leads to errors in identifying gradual expansion.

	Sample 01	Sample 02	Sample 03
Raw point cloud 0			


Object-based change prediction
Point-based change prediction
Change ground truth
Raw point cloud 1

Figure 29. Point cloud visualization of three sample regions. Rows by order represent point cloud at: epoch 0, epoch 1, ground truth, point-based results, and the object-based results. Red ellipses highlight key areas of interest discussed in the text.

The object-based method provides more stable results with reduced noise due to structured segmentation. However, the clustering influences the shape of the predicted changes, especially at object boundaries. When clusters do not match the exact shape of the change, the detection follows the cluster limits rather than the actual change boundary. This effect is noticeable along the edges of buildings and trees, where the segmentation structure affects classification. **Figure 30** presents the committed errors of both methods, showing the differences in how each approach manages change detection.

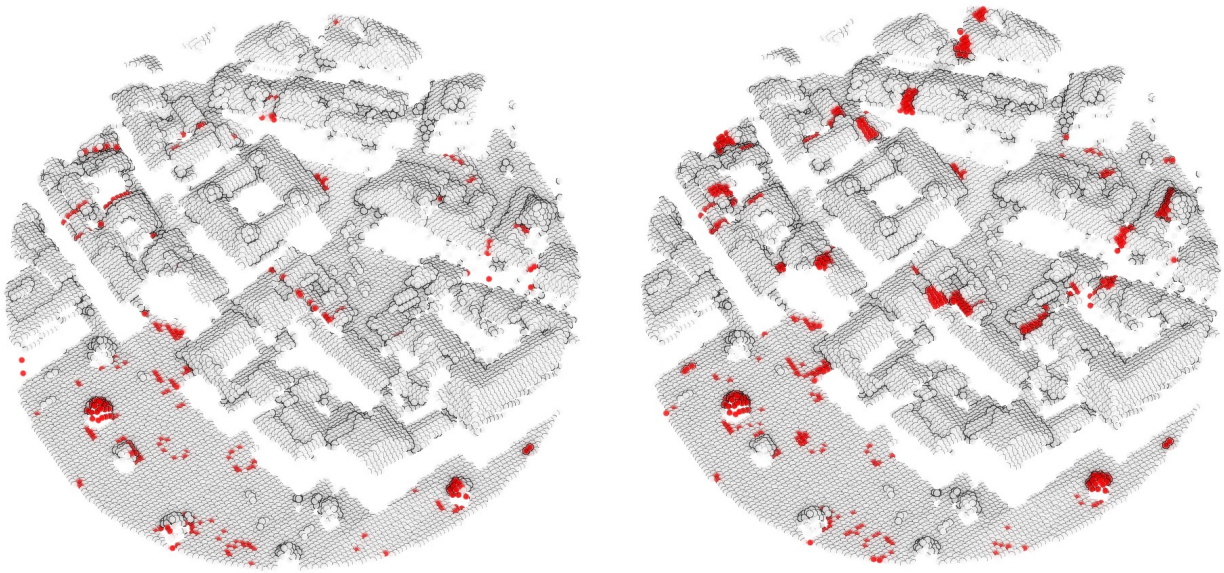


Figure 30. Error analysis in change classification in red color. Left: point based. Right: object based.

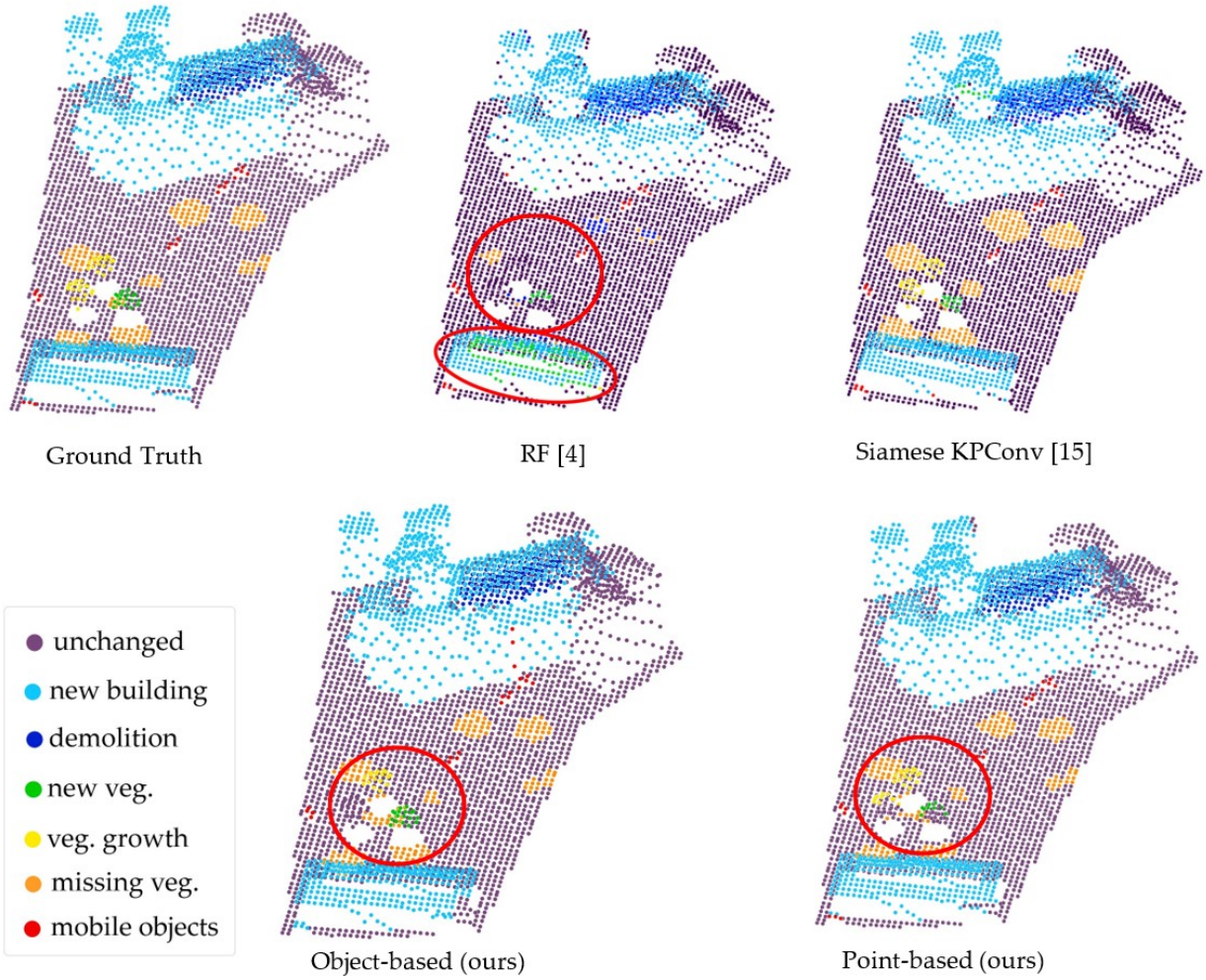


Figure 31. Visual comparative of our methods to random forest [130] and Siamese KPConv, problematic regions are highlighted by red circles.

2.4 Discussion

This study presents an object-based 3D change detection framework that integrates semantic segmentation and geometric feature analysis to improve classification accuracy and spatial consistency. Unlike point-based methods, which process individual points and are highly sensitive to noise, this approach groups points into spatially coherent objects, ensuring a more stable classification process. The framework segments the point cloud using the ℓ_0 cut-pursuit algorithm, matches objects across epochs, and classifies changes based on aggregated geometric and semantic features.

The results show that the method performs well in structured elements, particularly for new buildings (87.93% IoU) and demolitions (81.40% IoU), where segmentation helps maintain spatial coherence. However, vegetation-related changes remain more challenging, with 57.01% IoU for vegetation growth and 62.07% IoU for missing vegetation. These changes occur gradually and at finer scales, making them harder to track at the object level. The proposed method achieves high accuracy in mobile object classification because it learns associations from semantic labels in the previous epoch. Since mobile objects retain their class across epochs, the model automatically assigns them to the mobile object change class. When compared to state-of-the-art methods, the

object-based approach achieves a balance between accuracy, efficiency, and interpretability. Deep learning models, such as Triplet KPConv (84.08% mIoU), improve feature extraction but require large training datasets. Unsupervised methods, like DC3DCD EFSKPConv (57.06% mIoU), avoid labeling costs but are highly dependent on clustering quality. These comparisons highlight the trade-offs in 3D change detection, where accuracy, computational cost, and data availability influence model selection.

The results confirm that the object-based approach enhances spatial coherence while maintaining competitive classification performance. It reduces misclassifications in urban structures and ensures a structured classification process. However, limitations persist in tracking small-scale changes, where segmentation may smooth out finer details. Moreover, the nearest-neighbor object matching may misassociate objects in densely populated areas where multiple objects have close centroid positions. Future work could explore more advanced graph-based matching techniques or probabilistic models to improve correspondence accuracy. Another limitation lies in the reliance on pre-classified point clouds. While this ensures robust semantic integration, it restricts the method's applicability to datasets without prior classification. A potential improvement would be to integrate deep learning-based semantic segmentation within the pipeline, allowing for automated class inference. Additionally, while the object-based approach improves computational efficiency, it may oversimplify complex changes occurring at finer scales, as was demonstrated in the ablation study. Introducing adaptive segmentation techniques that dynamically refine clustering based on local geometry could enhance the framework's adaptability.

Future work should explore adaptive segmentation strategies to refine object boundaries, particularly for dynamic environments. Integrating feature-learning techniques from deep clustering [132], [174] could also enhance the method's ability to detect complex changes while maintaining interpretability and efficiency.

2.5 Conclusions

This chapter presents an object-based 3D change detection framework that integrates geometric and semantic fusion to improve classification robustness in LiDAR point clouds. The method maintains spatial coherence, particularly for structured elements such as buildings (96.57% IoU) and missing vegetation (57.01% IoU). However, segmentation may overlook small-scale changes, particularly in dynamic environments. The ablation study shows that a point-based approach achieves a higher mIoU (83.43%), with better performance in fine-grained categories such as vegetation growth (73.86%) and mobile object detection (100%). The lack of spatial constraints introduces more noise and classification inconsistencies, confirming a trade-off where object-based methods improve structural consistency, while point-based approaches capture finer details but remain sensitive to noise. Future work will explore hybrid models that combine adaptive clustering with attention-based architectures, such as SuperPoint Transformers, to refine object boundaries dynamically.

CHAPTER 03:

Semantic Segmentation for Change Detection in Urban Applications

Preface

Most 3D change detection methods output either point-level change maps or lists of changed objects, but structuring these changes into interoperable models is a newer area of interest. Change structuring refers to formalizing detected changes in a structured data model that can be easily interpreted, queried, and integrated into other systems. Instead of treating change detection output as an end, structured representation embeds changes within semantic 3D model of the environment. This is highly relevant to digital twin and city modeling applications, where one wants to maintain an up-to-date 3D city model with changes incorporated as new objects or updates.

In this third chapter, we address a constraint left open by the previous ones: ***How can detected changes be formalized into structured, interpretable, and interoperable models suitable for urban and railway applications?***

While the previous chapters focused on supervised workflows for semantic segmentation and object-based change detection, these approaches rely on annotated datasets, which are often costly to produce and difficult to generalize. In contrast, the method presented here operates without supervision, based on geometric reasoning and rule-based logic at the semantic object level. This offers a viable alternative in contexts where prior knowledge about expected change types is available, but annotated data is not.

The pipeline begins with a bi-temporal instance segmentation of point clouds to isolate buildings and vegetation. Changes are then inferred using predefined rules, based on object correspondence, surface distance, and thresholds, yielding categories such as appearance, disappearance, or modification. This object-centric strategy allows the method to remain interpretable while operating across different data sources.

The second part of the chapter addresses how change information can be structured. Rather than producing isolated outputs, changes are integrated into an extended CityJSON format, with a proposed extension that encodes temporal and semantic attributes. This step bridges the gap between raw detection and usable models, supporting scenarios like incremental 3D reconstruction, model updating, or archival.

This chapter reconnects with the initial motivations of the thesis: turning 3D change information into meaningful, structured representations usable in urban monitoring.

Based on the conference paper (Kharroubi et al. 2024)

Automated detection and structuration of building and vegetation changes from LiDAR point clouds

Kharroubi A, Ballouch Z, Jeddoub I, Hajji R, Billen R. *ISPRS Archives, XLVIII-2/W8-2024, 227–233.*

Abstract: Urban environments are continuously changing, driven by factors such as population growth and infrastructure expansion, which necessitates regular updates to urban models. Accurate, up-to-date information on these changes is critical, particularly for national mapping agencies monitoring long-term urban development. This chapter presents an automated methodology for detecting building and vegetation changes within urban environments using LiDAR point clouds, focusing on the city of Liège in Belgium. By leveraging recent aerial LiDAR data from 2022, our approach identifies, models, and integrates urban changes into a refined 3D Digital Twin model of Liège. The methodology includes preprocessing steps such as coordinate systems homogenization, noise filtering, and octree-based spatial indexing, followed by semantic and instance segmentation of point clouds using the RandLA-Net deep learning model. The change detection process focuses on four categories: appearance, disappearance, modification, and unchanged features. Achieving 100% accuracy for detecting new buildings changes, as validated within the study dataset and methodology. The modelled results are structured into a CityJSON city model. This automated approach significantly enhances urban model updates by integrating detected changes into a standardized 3D representation.

Keywords: 3D Change Detection, Semantic Segmentation, Building, Trees, LiDAR, CityJSON

3.1 Introduction

Urban areas undergo continuous transformations driven by population growth, expanding infrastructure, and increasing urbanization. Keeping urban models updated is vital for efficient planning, resource management, and environmental oversight. However, traditional methods for updating these models are labor-intensive, time-consuming, and often prone to errors. This highlights the need for efficient, automated techniques to monitor and integrate urban changes. LiDAR (Light Detection and Ranging) technology has become a key tool for urban modeling, providing high-resolution, three dimensional data that significantly improves the accuracy of urban model updates. Several studies have explored LiDAR's potential for change detection and model refinement. Extending LiDAR applications to 3D models has led to the development of semi-automated techniques focused on specific features like buildings [183] or vegetation [189]. Although promising, these methods have limitations: they typically address only one urban feature at a time, lack full automation, and struggle with scalability. Current research still faces key challenges. First, automating change detection across varied urban features remains complex due to the heterogeneity of urban landscapes, which include different building types, sizes, and vegetation densities. Second, most existing approaches struggle to integrate multiple urban elements, such as buildings and vegetation, into a unified model. Third, there is a need for scalable, cost-effective solutions that can be applied over large areas, as required by national mapping agencies and urban planners.

A critical component of our approach is the integration of semantic information with geometric changes to achieve comprehensive 3D semantic change segmentation. Below, we illustrate the difference between binary and semantic change detection. In binary change detection, changes are identified without differentiating the types of objects involved, making it difficult to extract meaningful urban insights. In contrast, semantic change detection provides a more nuanced analysis by distinguishing between different classes, such as buildings and trees. This distinction enables a better understanding of how urban features evolve and supports more informed decision-making.

To address the existing challenges, we present an automated workflow for detecting and modeling changes in urban buildings and vegetation using LiDAR point clouds. The research focuses on the city of Liège in Belgium, an area with a diverse urban fabric and ongoing development. We use recent aerial LiDAR data from 2022 to update the city's Digital Twin in Level of Detail (LoD) 2.2. The methodology involves several steps: preprocessing the data to remove noise and outliers, using deep learning models for semantic and instance segmentation, and detecting changes based on an object-based approach. Specifically, we employ RandLA-Net for semantic segmentation to classify the point cloud into buildings, vegetation, and other features. Changes are categorized into four types: appearance, disappearance, modification, and unchanged features. The results are structured into a CityJSON model using our proposed new change extension. Our approach provides a practical and efficient solution for urban model updates, benefiting regional and national authorities involved in urban monitoring and planning **Figure 32**.

The main contributions of this research are: (1) a fully automated workflow for urban change detection, (2) a new CityJSON extension for change information structuring and management.

The structure of this chapter is as follows: Section 2 reviews related work, highlighting the current state and limitations of existing methods. Section 3 describes our methodology, including data preprocessing, segmentation, and change detection. Section 4 presents the results and their validation. Finally, Section 5 and 6 concludes the study and discusses potential future research.

Figure 32. Moving from raw change detection results to actionable information in the context of an application.

3.2 Related works

3.2.1 Semantic segmentation

In the dynamic field of 3D point cloud semantic segmentation, significant advancements have been made through various approaches [190]. Model-driven methods, including RANSAC [191], Hough Transform, and region growing, continue to provide a solid foundation for segmenting point clouds, offering robustness in fitting geometric shapes and handling locally homogeneous properties. Meanwhile, knowledge-driven methods [192], [193], incorporating explicit rules and matching strategies, are often used to enhance segmentation results in tandem with other approaches. Rule-based algorithms follow a set of predefined rules to categorize points, offering simplicity and computational efficiency [194]. Matching-based methods, on the other hand, excel in accuracy and robustness by matching the point cloud to a database of 3D models. These methods are particularly useful in scenarios where precise object recognition is essential, but they can be computationally intensive and require extensive model databases. Knowledge-driven methods often complement data-driven and model-driven approaches by refining segmentation results, ensuring that the identified objects align with domain-specific knowledge or predefined rules.

Finally, data-driven methods [190] have gained popularity, using machine and deep learning techniques to extract features and classify points. These data-driven methods span projection-based strategies, including multi-view [195], [196], [197], [198], and spherical image approaches [199], [200], [201], as well as volumetric [202], [203], [204], [205] and lattice-based representations [206], [207], [208], with hybrid representations combining their strengths. Point-based methods have made significant strides, introducing neural network architectures like PointNet and PointNet++ [209], [210], point convolution networks [211], [212], [213], and Graph Neural Networks

[214], [215], [216] to capture complex relationships and dependencies within point clouds. Recently, transformer-based methods [174], [212], [217], [218], [219] have emerged as state-of-the-art, using attention mechanisms to model global dependencies in point clouds. **Figure 33** summarizes these semantic segmentation methods.

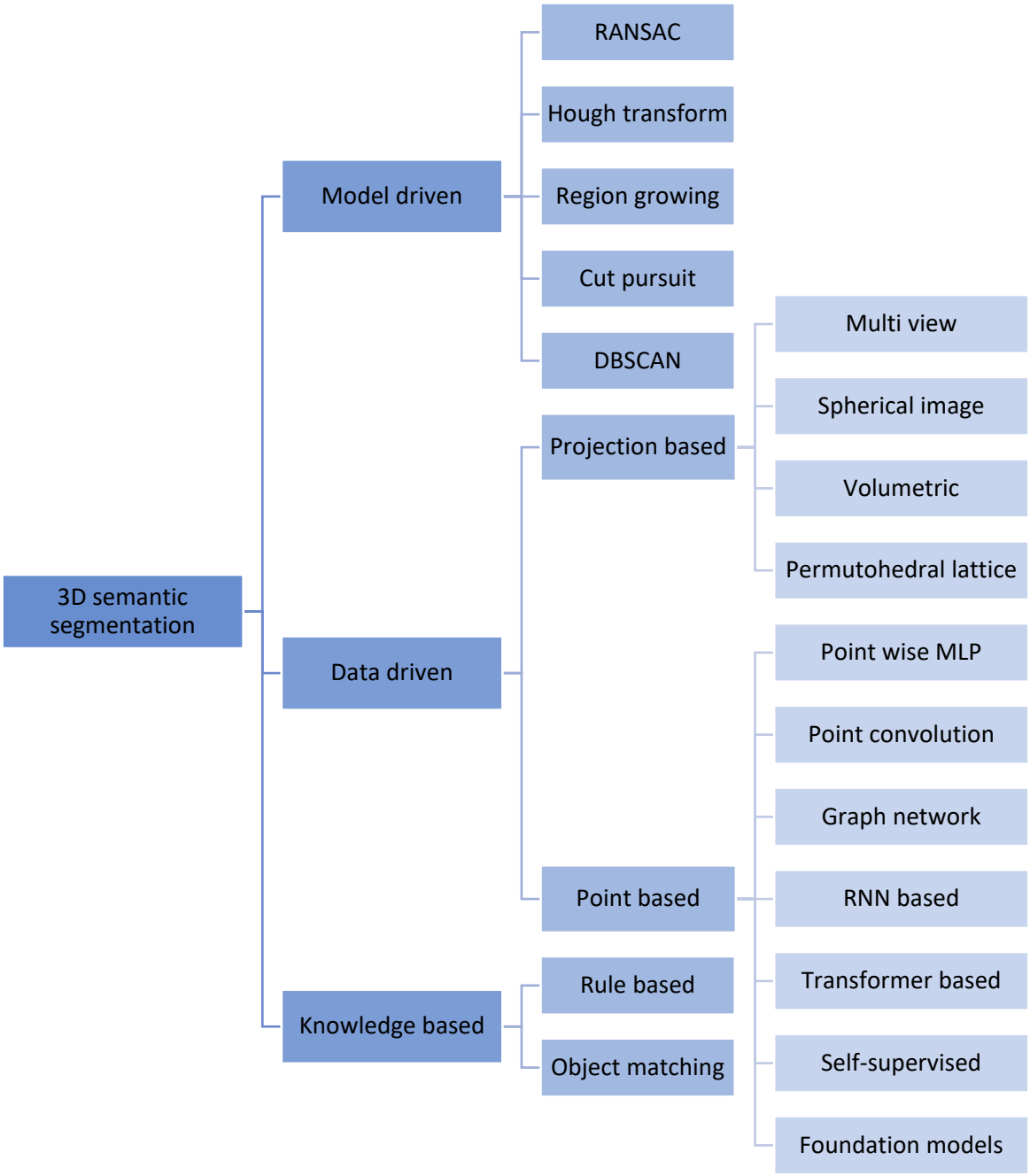


Figure 33. Summary of 3D semantic segmentation methods.

The most recent advances in 3D semantic segmentation include self-supervised learning, multi-task learning, transformer networks, foundation models, and 3D-LLMs. Self-supervised learning methods can train models without the need for labelled data [220]. This can be helpful for reducing the cost and effort of training 3D semantic segmentation models, especially for large-scale applications. Multi-task learning methods train models to perform multiple tasks simultaneously, such as object detection and

scene understanding [221]. This can help 3D semantic segmentation models to learn better representations of the data, which can lead to improved performance on all the tasks. Transformer networks are a type of neural network architecture that has been shown to be effective for 3D semantic segmentation [217]. They can learn long-range dependencies in point clouds, which can be helpful for better results. Foundation models, such as SAM [222], are Large Visual Models (LVMs) that have been trained on a massive dataset of images and text. Foundation models can be used to train 3D semantic segmentation models in a variety of ways, such as by training a downstream model on top of the foundation model. Finally, the 3D-LLMs [223] is a new type of large language model that is specifically designed to understand and reason 3D data. 3D-LLMs can be used to perform a variety of 3D-related tasks, such as 3D captioning, 3D questions answering, and 3D-assisted dialog.

3.2.2 Change detection

3D point clouds have become a crucial data source for monitoring vegetation and buildings in urban environments. Compared to traditional image data, 3D points clouds provide richer information by capturing the geometric and structural details of objects, facilitating more accurate monitoring and analysis [224]. However, the irregular and unstructured nature of point clouds present unique challenges, requiring advanced methods to extract meaningful insights. Change detection approaches can be broadly categorized into two main families: derived-product methods and direct comparison techniques.

Approaches based on derived products: Derived-product methods transform raw 3D point clouds into structured representations like Digital Surface Models (DSMs) or voxel grids, making it easier to apply traditional 2D change detection algorithms. For example, DSM-based methods enable change detection in buildings by calculating height differences between multi-temporal datasets [225]. Similarly, Canopy Height Models (CHMs) derived from LiDAR data are used to monitor vegetation changes, with techniques like watershed algorithms identifying individual tree crowns [226]. Voxel-based methods also regularize point clouds for easier comparison, reducing the impact of varying point densities [49]. However, these derived-product methods have inherent limitations. The conversion from 3D to 2D or voxelized grids often results in information loss and interpolation errors, which can reduce the accuracy of change detection, particularly for objects with fine geometric details. Moreover, studies have shown that derived approaches struggle in areas with complex topography or where high precision is required, making direct comparison methods more suitable for such scenarios [227].

Direct approaches using raw point clouds: Direct methods operate on raw 3D point clouds, preserving the spatial and topographical relationships of data points. Traditional techniques like Cloud-to-Cloud (C2C) and Multi-Scale Model-to-Model Cloud Comparison (M3C2) measure distances between points in multi-temporal datasets to detect changes [43], [228]. While effective for precise geometric comparisons, C2C methods often fail to provide context about the type of changes, such as the appearance or disappearance of buildings or vegetation. Recent advancements have incorporated semantic segmentation into change detection workflows, referred to as post-classification change detection. For instance, [229] introduced a connected component

analysis method for updating building information in topographic maps, while [230] developed an object-based approach that detects changes like newly constructed or demolished buildings. [130] extended this to vegetation monitoring, identifying changes at the tree level using a supervised classification framework.

Machine learning and deep Learning have significantly enhanced the accuracy of 3D change detection. Deep learning models, including Siamese Neural Networks and Graph Convolutional Networks (GCNs), have introduced more advanced solutions, processing raw point clouds directly and achieving high performance in change detection task [169]. Siamese KPConv, a notable deep learning approach, directly operates on 3D point clouds to detect changes, reducing the need for voxelization and retaining geometric details [231]. However, deep learning models often require substantial computational resources and large annotated datasets, which can be a limitation in real-world applications. Self-supervised learning techniques are emerging to mitigate these data challenges, employing deep clustering and contrastive learning to improve the performance of unsupervised 3D change detection methods.

Semantic Change Detection (SCD) extends traditional methods by identifying both the regions and types of changes, such as distinguishing between a changed building and a tree. SCD approaches can be categorized into single encoder, dual encoder, and triple encoder models. Single encoder methods often suffer from class overlaps, while dual encoder models like Siamese architecture provide better differentiation but may miss temporal relationships between features. Triple encoder approaches incorporate auxiliary information, enhancing accuracy but increasing computational complexity.

Despite advancements in geometric and semantic change detection, current methods often focus on one aspect, neglecting the other, reliance on labeled datasets, and limited scalability. This study introduces an unsupervised semantic change detection (SCD) pipeline that integrates both semantic and geometric analysis using 3D point cloud.

3.3 Methodology

This study presents a comprehensive methodology for automating the detection and modeling of building and vegetation changes using LiDAR point clouds. As summarized in **Figure 34**, the workflow begins with data preparation, including coordinate systems standardization, cleaning, and preprocessing of bi-temporal datasets to ensure consistency. Semantic and instance segmentation are then performed to accurately classify and isolate objects such as buildings and trees. The subsequent steps involve defining and detecting changes, quantifying these changes using relevant metrics, and generating detailed 3D models. Finally, the results are structured into the CityJSON format, facilitating seamless integration into urban digital twin applications.

3.3.1 Data preparation

Before the implementation phase, several data preprocessing steps were conducted. First, adaptation to the same coordinate system was performed. This step was followed by cleaning and merging the tiles covering the study area. Finally, the point clouds were prepared according to the requirements of the deep learning architecture used for classification.

3.3.2 Semantic and instance segmentation

Semantic segmentation was performed to accurately extract buildings and trees. To perform semantic segmentation, we used RandLA-Net as a deep learning model, which had been trained and validated on a dataset comprising urban LiDAR scans from the city of Liège, as described in our earlier work [232]. This model achieved classification accuracy of 95% for buildings and vegetation, ensuring reliable semantic segmentation in this study. Following this, instance segmentation was applied to trees using the TreeISO algorithm [233] to isolate each tree individually and assign a unique identifier. TreeISO segments individual trees using a hierarchical, graph-based clustering approach. It first groups points into fine clusters based on spatial proximity, then aggregates these into tree's segments using connectivity rules informed by elevation and distance. While designed for terrestrial laser scanning (TLS), we adapted it for aerial laser scanning (ALS) to isolate individual trees effectively from our dataset. For buildings, instance segmentation was not required, as modeling relied on PICC data, which includes footprints and unique identifiers for each building.

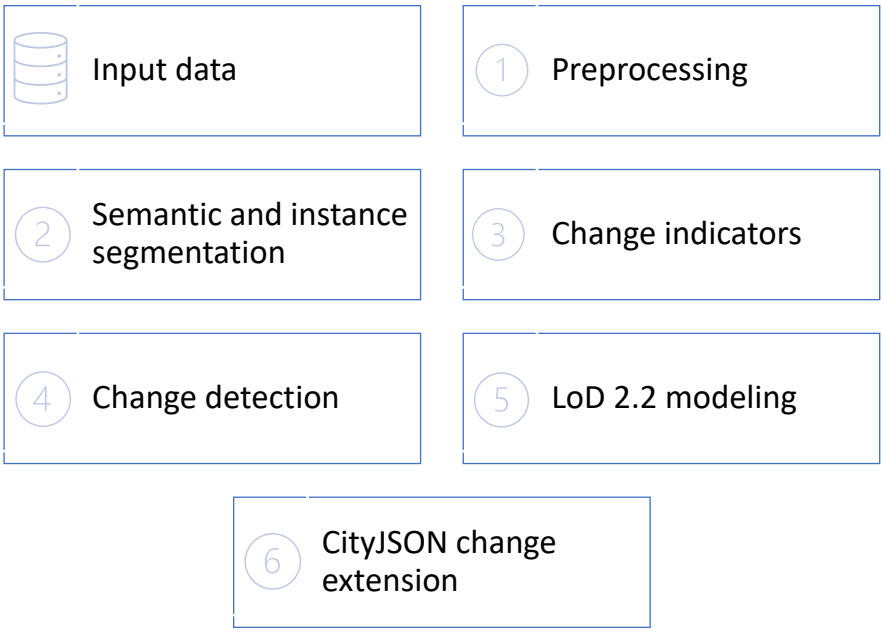


Figure 34. Overview of the Methodology Workflow.

3.3.3 Change definition and detection

To detect changes between two epochs, we propose a method that combines point-level indicators, clustering, and object-level metrics. This approach ensures robust identification of changes while addressing common artifacts, such as façade-related noise in buildings and overlapping canopies in trees. Changes are categorized into four classes for both buildings and trees: new, lost, modified, and unchanged.

Point-level change detection begins with the calculation of distances between the two epochs using a modified version of the Multi-Scale Model-to-Model Cloud Comparison (M3C2) algorithm. To mitigate artifacts from vertical structures like building façades or dense tree canopies, M3C2 distances are projected onto a horizontal plane. This ensures

that the detected changes reflect meaningful modifications in object presence or geometry rather than irrelevant vertical differences caused by variations in scan viewpoints (see **Figure 42**). Using these distances, objects are initially classified as new or lost based on horizontal threshold:

- New objects are detected in the second epoch where M3C2 distances exceed a threshold, indicating objects present in the second epoch but not in the first.
- Lost objects are identified from the first epoch where M3C2 distances exceed the threshold, signifying objects present in the first epoch but absent in the second.

To refine these classifications, a connected-component clustering algorithm is applied. This groups adjacent points with similar change indicators, forming spatially coherent regions of change. Clusters that fall below minimum size, typically caused by noise or minor misalignments, are removed to ensure that only meaningful changes are retained. For objects classified as unchanged, vertical changes are further analyzed. Buildings are examined for structural modifications such as height increases (e.g., additional floors) or reductions (e.g., partial demolitions), while trees are assessed for vertical growth or trimming. This vertical analysis leverages the M3C2 distances in the z-dimension to detect significant differences and our metrics for change quantification.

3.3.4 Object change metrics

After detecting changes, the next step involves quantifying them to characterize the modifications for both buildings and trees. These metrics provide detailed insights into the type and extent of changes.

For modified buildings, the following metrics are extracted:

- Height Difference: The average and maximum height differences are calculated to identify taller or shorter structures.

For trees, we detect four categories of change: new, lost, modified, and unchanged. New, lost, and unchanged trees are identified using M3C2 distance metrics to compare spatial correspondence between epochs. For modified trees, the following metrics are used to assess growth or pruning:

- Tree Height (99th Percentile Z): The maximum height of the tree above ground. This metric identifies the maximum canopy height while avoiding influence from outliers (e.g., single, unconnected high points due to noise).

These metrics provide a clear and quantifiable basis for evaluating structural changes in individual trees. Since trunk positions are unavailable due to the aerial acquisition, we identify corresponding tree objects between epochs by finding the closest tree object spatially. If the nearest tree surpasses a defined maximum distance, the tree is flagged for manual user confirmation. This assumes new and lost trees have already been filtered out, leaving modified or unchanged trees for comparison. For each tree in the second epoch, the nearest tree in the first epoch is determined based only on the spatial proximity of their centroids or canopy centers, without relying on constraints like canopy dimensions or height metrics, as these can vary due to changes.

Changes are then classified using either threshold-based rules or supervised learning techniques if a labelled dataset is available. Threshold-based rules are predefined, such as classifying trees with height increases above a certain value as "heightened" or "lowered". Alternatively, when labeled training data is available, machine learning algorithms like Random Forests can classify changes based on the extracted metrics. This approach allows for more nuanced classification while leveraging a training dataset.

3.3.5 Buildings and trees modelling

After the change detection step, 3D modeling of different change classes is conducted. For buildings, the footprints extracted from the previously described PICC data are used. They include attribute information for each building, along with unique identifiers. In addition to these vector data, building point clouds, derived from semantic segmentation, were used to reconstruct building geometry. Geoflow tool[3] enabled this reconstruction in CityJSON format, achieving a level of detail 2.2. For the 3D modeling of trees, the instance segmentation results were used to calculate the height of each tree above the ground, as well as specific parameters (height, width, perimeter, etc.). These extracted parameters were also reconstructed in CityJSON format using a Python script. The schema and geometry of the reconstructed models were then validated. For geometry validation, we used Val3dity[4], an open-source software dedicated to validating 3D primitives (geometries) of the model. For schema validation, we relied on the official validator[5] for CityJSON files.

3.3.6 Structuring into CityJSON

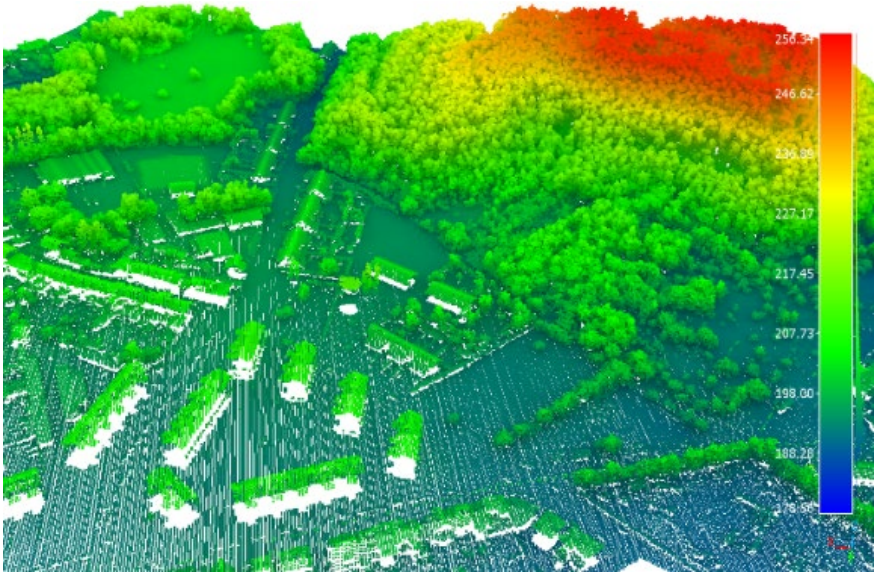
This step involves the structuring of the change's information into a standardized data model to enhance usability and interoperability. CityJSON is a lightweight and developer-friendly format for representing, storing, and extending 3D city models. Given its simplicity and its extensibility, we integrate the change information in compliance with the CityJSON standard by defining new attributes for existing city objects, namely buildings and vegetation. We define a set of attributes that report the change characteristics (e.g., the change type (refer to **Table 11**), the height variation, and the change uncertainty). We structure this information following the CityJSON extension specifications. The change extension is adopted upstream as a way of organizing, storing, and managing the change information in 3D city models. Following the CityJSON extension specifications 2.0.1, the change attributes are integrated as extraAttributes for existing city objects. Each city object is extended by the three main change attributes: change, height variation, and change uncertainty. Their values are stored respectively following the defined CityJSON extension.

[3] https://github.com/geoflow3d/geoflow-bundle
[4] https://github.com/tudelft3d/val3dity
[5] https://validator.cityjson.org/

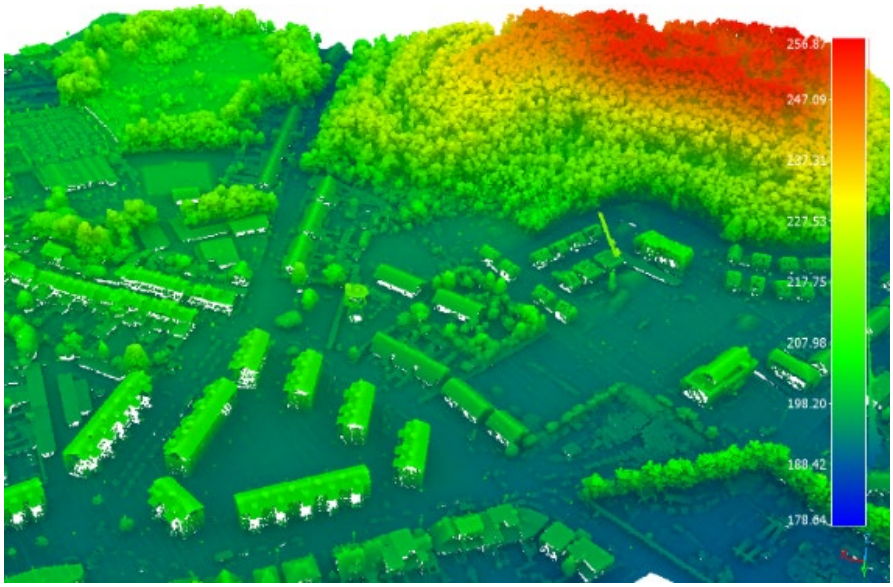
3.4 Experiments and results

3.4.1 Data description

In this study, two types of data were used: LiDAR and vector. The LiDAR point clouds from 2022 have specific characteristics, including an average flight altitude (AGL) of 2400 m, a density of 6.8 points/m^2, and the use of Double LMSQ780 and Double VQ780II-S equipment (**Figure 35**. The planimetric accuracy is evaluated with an RMSE of ≤ 1 m, while the altimetric accuracy reaches an RMSE of ≤ 0.4 m. In contrast, the point clouds from 2014 were collected at a minimum altitude of 1015 m and a maximum altitude of 1550 m, with a density of only 0.8 pulses/m^2, resulting in an average spacing of 1.13 m in the direction of flight and 1.14 m transversely, while also displaying a planimetric and altimetric accuracy with a maximum RMSE of 1 m and 0.4 m, respectively. Additionally, data from the Mapping Framework Information Plan (PICC) were utilized; these vector data allowed for the extraction of footprints and attribute information for each building, complementing the analysis derived from the point clouds.



(a). LiDAR acquisition of 2014



(b). LiDAR acquisition of 2022

Figure 35. (a). Epoch 1 and (b). 2 of aerial point clouds over the study area colored by Height ramp.

3.4.2 Implementation

Semantic segmentation was performed using RandLA-Net, a deep learning model previously validated for urban classification tasks. The model classified the point cloud into four classes: buildings, vegetation, ground, and others. The overall accuracy for buildings and vegetation exceeded 95%. The experiments were conducted on a high-performance workstation equipped with an NVIDIA GeForce RTX 3090 GPU, an Intel i9-10980XE CPU, and 256 GB of RAM.

After semantic segmentation (**Figure 36**), vegetation classes were further processed using TreeISO for instance segmentation. However, due to the inclusion of fences and low vegetation in the vegetation class, additional filtering was applied (**Figure 37**). To isolate valid tree clusters and exclude noise or irrelevant objects (e.g., fences or isolated points), we applied a filtering criterion based on the height distribution of points within each cluster. For each segmented cluster, the 95th percentile of point heights was calculated. Clusters where the 95th percentile height exceeded 3.5 m were retained, as this threshold reflects a meaningful minimum height for trees in the study area. This percentile-based approach ensures robustness by adapting to variations in point density and cluster size, reducing sensitivity to noise and outliers compared to a fixed point count.

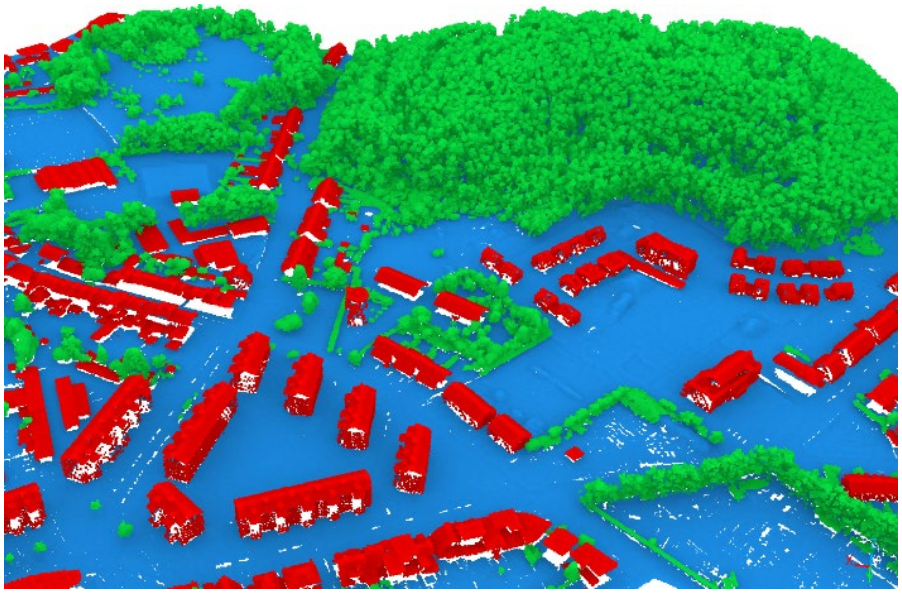


Figure 36. Results of semantic segmentation of 2022's data.

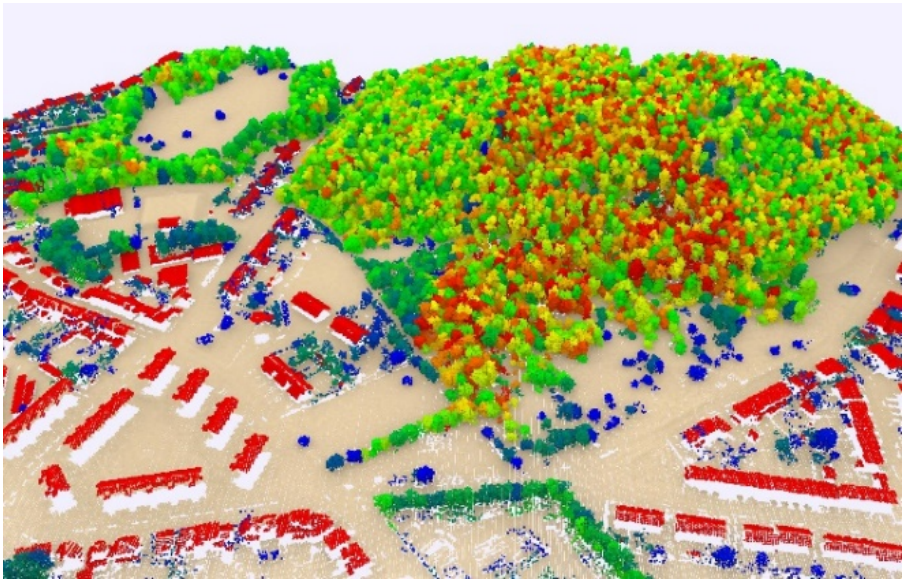


Figure 37. Vegetation clustering using TreeISO after removing small components, low vegetation, and fences (data 2014).

To detect new and demolished buildings, the building class was projected onto the XY plane, and the M3C2 algorithm was applied (**Figure 38**). For new buildings, epoch 02 served as the reference, while epoch 01 was inverted to identify demolished buildings. Height changes were calculated for individual buildings, which were isolated using PICC shapefile footprints. We applied a height threshold of 3 meters, equivalent to the average height of a floor, to detect significant vertical changes. Buildings with height increases exceeding 3 meters were classified as "heightened," while those with reductions exceeding 3 meters were classified as "lowered." However, no such vertical changes were detected in the study area, as all buildings remained within this threshold or were completely new.

Figure 38. The new building visualized in green, and unchanged in gray.

For trees, we applied a height threshold of 1 meter to detect significant vertical changes, considering it sufficient to capture meaningful growth or pruning while avoiding noise from minor seasonal variations or data inaccuracies. Trees with height increases exceeding 1 meter were classified as "grown," while those with reductions beyond this threshold were classified as "pruned." However, challenges arose in areas with dense canopy coverage, where overlapping layers occasionally affected the clustering process, leading to misclassification (**Figure 39**). This limitation emphasizes the importance of refining segmentation techniques to handle complex vegetation structures more effectively in future studies.

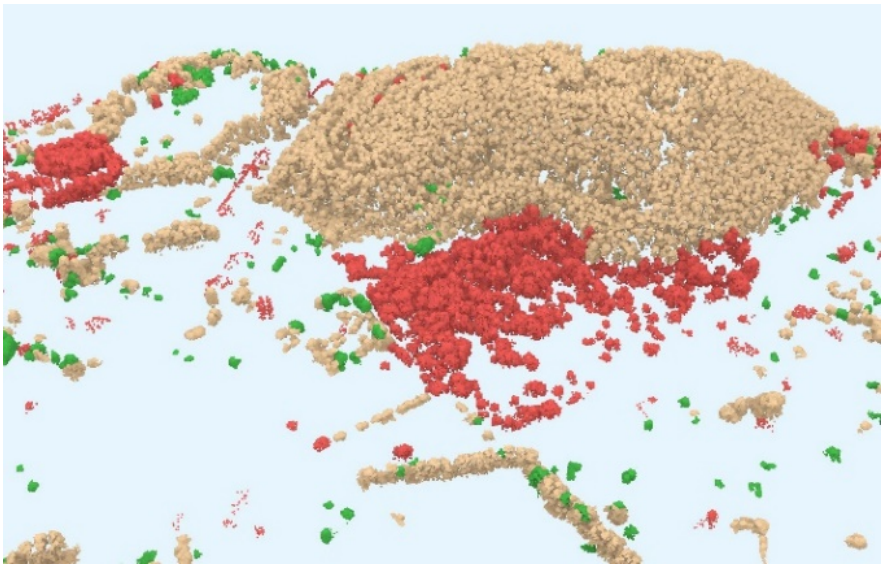


Figure 39. New vegetation visualized in green, removed in red and modified in brown.

In this work, change attributes were not included in the reconstruction process, as described in Section 3.5. We have therefore proceeded as follows: for each city model, we upgraded the city models v1.1 for buildings and v1.0 for vegetation to version v2.0

using CJIO to comply with our extension, and we have added the corresponding attributes using Python code. For instance, in the LoD2.2 building model, representing all new buildings, we have assigned the change attributes to each building, reflecting the type of change (i.e., new), the height variation, and the change uncertainty. We proceeded in the same way for vegetation to report the change type.

Table 11. Change type according to the city objects.

City Objects	Building	Vegetation
Change type	Unchanged, New, Demolished, Heightened, Lowered	Unchanged, New, Lost, Growth, Trimmed

The CityJSON files containing the change attributes are validated using the CityJSON schema validator (**Figure 41**). The aim behind the change extension is to ensure that these attributes are maintained in a standardized way and are automatically updated when a new value is available. Results are viewed below in **Figure 40**.

Figure 40. The results are displayed in an in-house developed tool for 3D city model structuration, management, and visualization [234], ©GeoScITY.

3.5 Discussion

The performance of the proposed methodology was qualitatively assessed by visually inspecting the results for new, lost, modified, and unchanged buildings and vegetation. Visual comparisons between the two epochs demonstrated that all new and demolished

buildings were correctly identified. The use of the M3C2 algorithm, combined with the projection onto the XY plane, minimized false positives associated with vertical structures, such as façades or overlapping tree canopies. This ensured reliable detection of changes in urban environments.

While the methodology accurately detected new and lost trees, some errors were observed in the classification and clustering stages. Specifically, small clusters of vegetation were occasionally misclassified as trees, and dense vegetation often led to the over-segmentation of tree canopies. These limitations highlight the need for further refinement in the instance segmentation and clustering algorithms. Additionally, the proposed methodology relies on several threshold values (e.g., the height threshold for tree filtering and the M3C2 distance threshold for change detection). These thresholds were determined empirically and may affect the reproducibility and generalizability of the results. Future work will explore adaptive or data-driven approaches to optimize these thresholds across different datasets.

Overall, the methodology demonstrates robustness in detecting major urban changes, particularly for buildings. However, the challenges associated with vegetation segmentation and classification suggest areas for improvement in future research.

3.6 Conclusions

This chapter introduced an automated methodology for detecting and structuring changes in buildings and vegetation using bi-temporal LiDAR point clouds. By combining semantic segmentation, object-based clustering, and geometric analysis, the approach effectively identified new, lost, modified, and unchanged features, structuring them into a CityJSON format for urban digital twins. Applied to LiDAR data from Liège, Belgium, the method demonstrated high reliability in building detection and vegetation, leveraging tailored metrics like tree height. However, challenges remain in accurately clustering dense vegetation and reduce the reliance on empirically derived thresholds. Future work will focus on improving threshold adaptability, integrating additional urban features, and exploring supervised learning for enhanced scalability and robustness in large-scale urban monitoring.

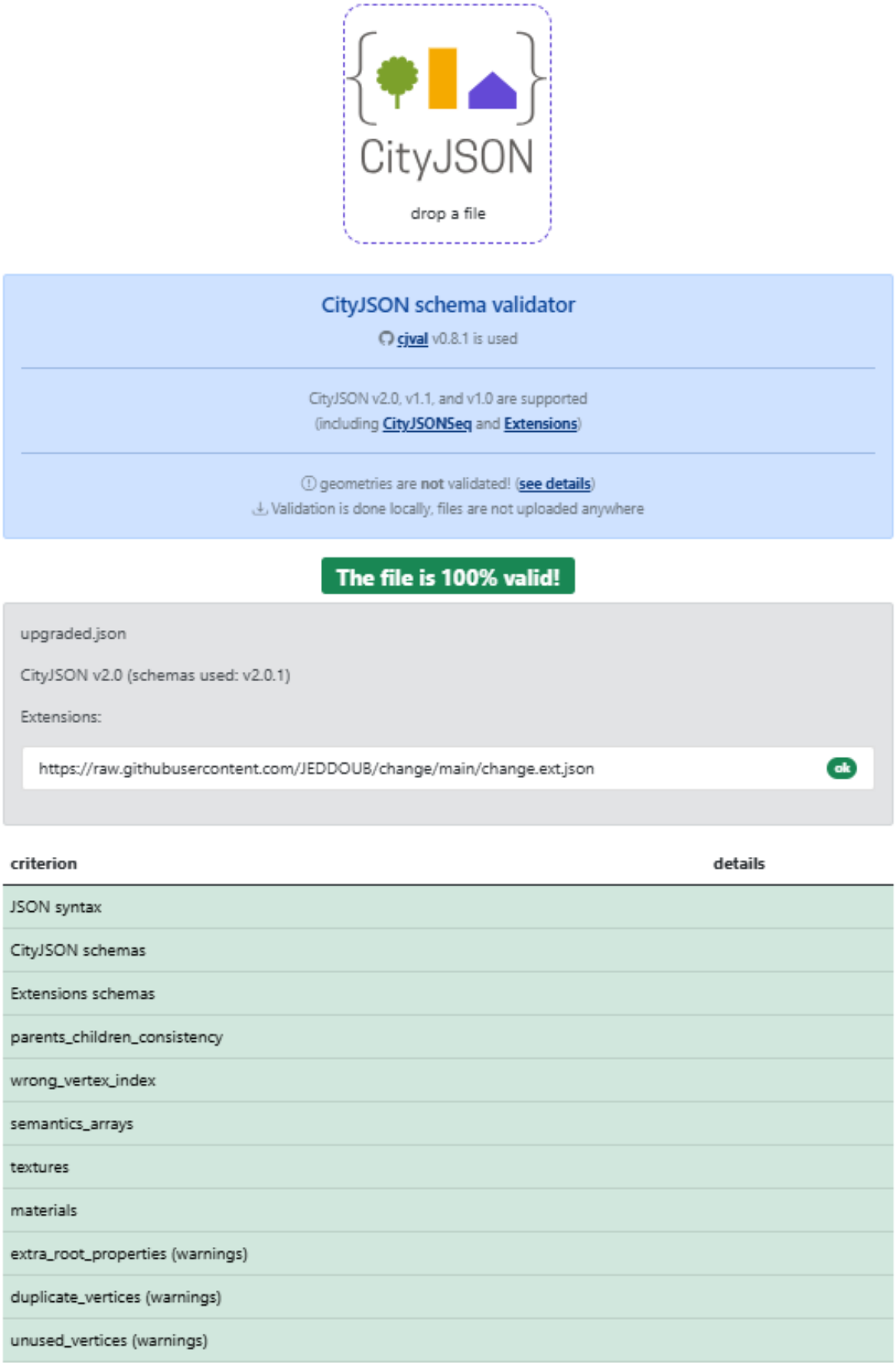
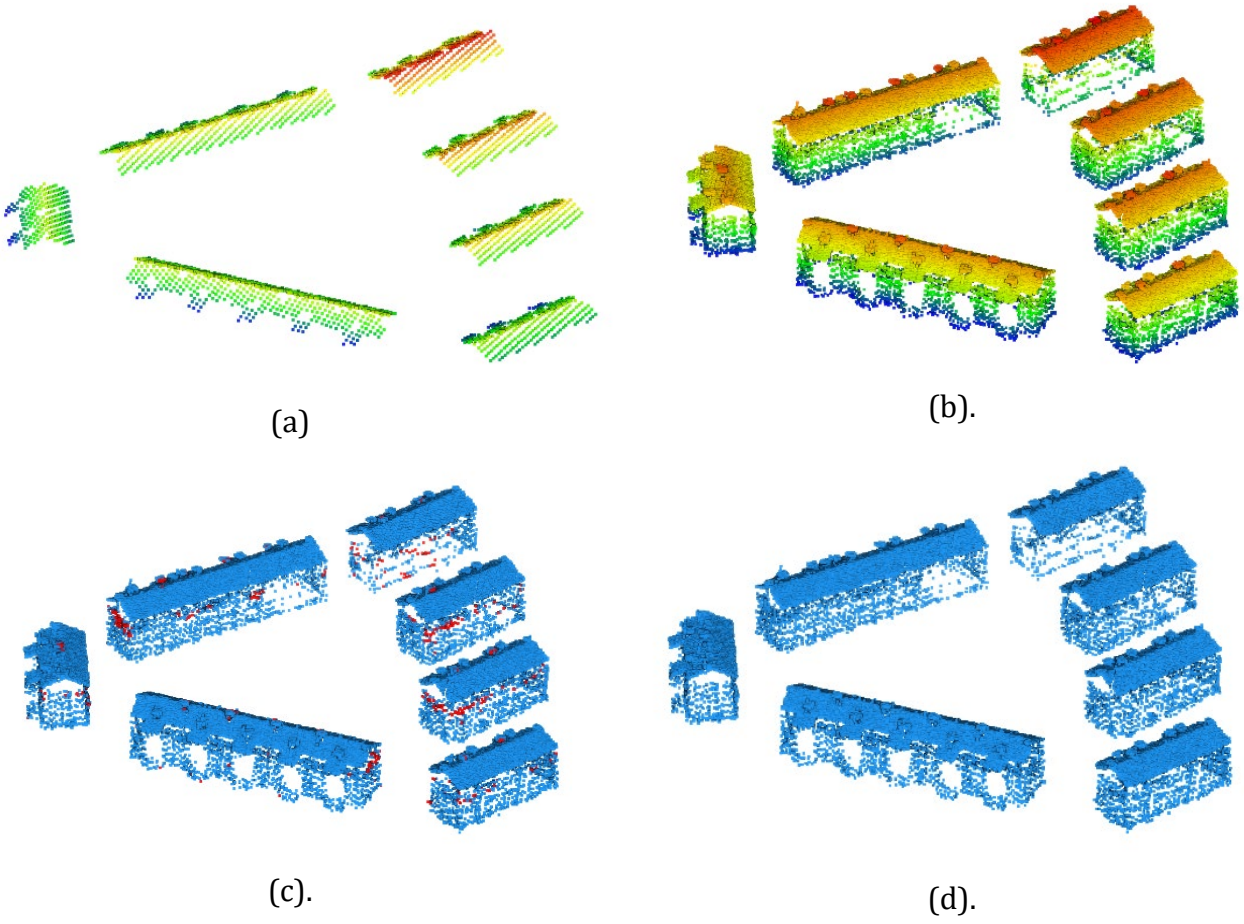


Figure 41. Change extension schema validation.



(a) (b). (c). (d).

Figure 42. Visual Comparison of LiDAR point clouds and change detection Results: (a) Point Cloud from 2014, (b) Point Cloud from 2022, (c) Significant Changes Detected Using M3C2 (Red Indicating Façade Noise as relevant change), and (d) Noise-Free Changes Using Projected M3C2.

Below is the created changed extension, also available at: https://github.com/JEDDOUB/change/blob/main/change.ext.json


```
{
 "type": "CityJSONExtension",
 "name": "changedetection",
 "url": "https://raw.githubusercontent.com/JEDDOUB/change/main/change.ext.json",
 "version": "0.5",
 "versionCityJSON": "2.0",
 "description": "Extension for structring change in 3D city models",
 "extraAttributes": {
     "Building": {
      "+change": {
       "value": "string"
      },
      "+height-variation": {
       "value": "number"
      },
      "+change-uncertainty": {
       "value": "number"
      }
      },
     "BuildingPart": {
      "+change": {
       "value": "string"
      },
      "+height-variation": {
       "value": "number"
      },
      "+change-uncertainty": {
       "value": "number"
      }
      },
     "SolitaryVegetationObject": {
      "+change": {
       "value": "string"
      },
      "+height-variation": {
       "value": "number"
      },
      "+change-uncertainty": {
       "value": "number"
      }
     },
     "Road": {
      "+change": {
       "value": "string"
      },
      "+height-variation": {
       "value": "number"
      },
      "+change-uncertainty": {
       "value": "number"
      }
     }
   },
  "extraCityObjects": {},
  "extraRootProperties": {},
  "extraSemanticSurfaces": {}
```

CHAPTER 04:

Semantic Segmentation for Railway Change Detection

Preface

Chapter 03 introduced an object-based change detection workflow that uses semantic segmentation to identify 3D changes. While the pipeline was evaluated on urban scenes, its application in structured domains, such as transportation networks, requires adaptation. This chapter focuses on railway environments and examines how semantic segmentation can support vegetation monitoring using point clouds acquired by mobile mapping systems.

Railway corridors represent complex linear systems where vegetation presence and growth are subject to strict regulatory constraints. The integration of dense 3D data allows for continuous monitoring, but interpretation at scale remains difficult due to the variability in acquisition conditions, lack of annotated data, and the progressive nature of vegetation change. This chapter responds to the third research sub-question from a practical viewpoint: ***How can detected changes be formalized in a structured and interpretable manner for railway applications?***

To address this, we construct and evaluate semantic segmentation models based on multi-context mobile mapping data. The Rail3D dataset is introduced as part of this work, including diverse scenes from multiple countries and annotated classes relevant to railway analysis. The evaluation focuses on supervised classification performance using point-based and ensemble models. In parallel, a section is dedicated to the analysis of vegetation change across time by applying the semantic segmentation models to bi-temporal acquisitions. This integration provides a preliminary but functional basis for detecting vegetation change and clearance along the tracks.

Based on the article (Kharroubi et al. 2024) with additional work on vegetation change detection

Multi-Context Point Cloud Dataset and Machine Learning for Railway Semantic Segmentation

Kharroubi A, Ballouch Z, Hajji R, Yarroudh A, Billen R. *Infrastructures. 2024; 9(4):71.*

Abstract: Railway scene understanding is crucial for various applications, including autonomous trains, digital twining, and infrastructure change monitoring. However, the development of the latter is constrained by the lack of annotated datasets and limitations of existing algorithms. To address this challenge, we present Rail3D, the first comprehensive dataset for semantic segmentation in railway environments with a comparative analysis. Rail3D encompasses three distinct railway contexts from Hungary, France, Netherlands, and Belgium, capturing a wide range of railway assets and conditions. With over 288 million annotated points, Rail3D surpasses existing datasets in size and diversity, enabling the training of generalizable machine learning models. We conducted a generic classification with nine universal classes (Ground, Vegetation, Rail, Poles, Wires, Signals, Fence, Installation, and Building) and evaluated the performance of three state-of-the-art models: KPConv (Kernel Point Convolution), LightGBM, and Random Forest. The best performing model, a fine-tuned KPConv, achieved a mean Intersection over Union (mIoU) of 86%. While the LightGBM-based method achieved a mIoU of 71%, outperforming Random Forest. This study will benefit infrastructure experts and railway researchers by providing a comprehensive dataset and benchmarks for 3D semantic segmentation. The data and code are publicly available for France, Netherlands, and Hungary, with continuous updates based on user feedback.

In addition, we apply the segmentation results to detect change using bi-temporal point clouds acquired over railway networks in Netherlands. Particular attention is given to vegetation, due to its operational impact on visibility and safety. Vegetation changes are quantified through object-based comparison of semantic classes across epochs, allowing for the identification of zones with changes or stable presence. The results confirm the importance of accurate semantic predictions as a prerequisite for reliable change detection in linear infrastructure environments.

Keywords: railways; LiDAR; dataset; semantic segmentation; machine learning, change detection

4.1 Context

Railway infrastructure plays a pivotal role in modern transportation systems, necessitating continuous innovation and precise management and maintenance [235]. The integration of advanced technologies, such as mobile LiDAR (Light Detection and Ranging), into railway applications has introduced transformative capabilities to the industry. To fully leverage these technologies, data processing methods, like semantic segmentation, are essential for understanding the complex and dynamic railway environment. Semantic segmentation, a fundamental component of computer vision, enables the categorization of each pixel in an image or point in a point cloud into predefined classes, facilitating scene understanding. In this context, machine learning has emerged as a promising approach. Deep neural networks, known for their ability to extract intricate patterns and representations from extensive datasets, offer the potential to automate critical tasks within the railway domain [236], [237].

However, the effective application of machine learning in the railway environment depends on a crucial factor: the availability of annotated data. Unlike urban contexts, where several annotated datasets and benchmarks are available [238], [239], [240], the railway environment lacks comprehensive benchmarks and annotated datasets tailored specifically to railway applications. The lack of annotated data presents a significant challenge for machine learning models development and evaluation. To train and validate semantic segmentation models effectively, labelled data, representing the diverse range of objects and structures found along railway tracks, including tracks, signals, poles, overhead wires, vegetation, and more, are required [241], [242].

Moreover, as railways turn to machine learning (ML), there is a rising need for models that are accurate and easy to interpret. Using complex models, especially in safety-critical areas like railways, raises concerns about interpretation. Traditional ML models are vital for building trust, especially in tasks like semantic segmentation, in the railway environment. Additionally, the challenge extends beyond data labelling; it also involves the establishment of benchmarks that enable researchers and practitioners to objectively assess model performance. Benchmarks serve as standardized metrics for the quantitative evaluation and comparison of novel algorithms. In the absence of a railway-specific benchmark, the assessment of model performance becomes subjective and ad hoc, hindering progress and the adoption of machine learning solutions in railway applications.

In response to this challenge, we introduce the Rail3D dataset, which aims to address the shortage of annotated data specifically tailored to railway applications. Rail3D comprises extensive point cloud data collected across diverse railway contexts in Hungary, France, Netherlands, and Belgium, covering approximately 5.8 km. This dataset not only mitigates the lack of annotated data but also serves as a foundational benchmark for semantic segmentation. The public availability of Rail3D represents a significant step toward fostering collaboration and innovation in the railway industry. We invite re-searchers, practitioners, and enthusiasts to explore, enhance, and contribute to this dataset, advancing the development of deep learning solutions that

enhance the safety, efficiency, and sustainability of railway infrastructure on a global scale.

In the railway sector, where decision-makers might not be machine learning experts, the clarity and reliability of models are paramount [242]. Benchmarking LightGBM, Random Forest, and the deep-learning-based KPConv allows us to assess the effectiveness of both traditional machine learning and advanced deep learning techniques. This comparative analysis is vital for selecting a model that not only delivers high performance but is also interpretable and aligns with safety-critical standards in railway applications. Through this benchmarking process, we aim to identify a model that combines accuracy and efficiency, ensuring it can be effectively used and trusted in the context of railway infrastructure management.

In addition to semantic segmentation, this chapter investigates the use of labelled bi-temporal data to support targeted change analysis, with a focus on vegetation. Vegetation along tracks poses operational constraints and is subject to regular maintenance. By identifying semantic classes at object level across epochs, we analyze spatial differences in vegetation presence and extent. The approach allows for the detection of change events based on semantic predictions. This method is integrated within the segmentation workflow and supports applications requiring monitoring of linear infrastructure over time.

The main objective of this chapter is to develop a large scale and multi-context 3D point cloud dataset for railway semantic segmentation tasks and assess models' performance on it. The contributions of our work are as follows:

- Present a multi-context railway point cloud dataset for semantic segmentation.
- Evaluate three state-of-the-art semantic segmentation models.
- Assess the generalization capabilities of the best performing model.
- Implement object-based vegetation monitoring using bi-temporal annotated data.

The rest of the chapter is organized as follows: Section 2 presents current change detection and semantic segmentation techniques, with a special focus on railway applications and available datasets. Section 3 outlines the specifics of the study area, the subset we curated to develop Rail3D, explains how we annotated the data, and introduces the three baseline methods selected for our analysis. Section 4 details the experiments we conducted and their outcomes. Sections 5 and 6 discusses what we found and ideas for future research.

4.2 Related works

The availability of datasets for training has so far played a key role in the progress and comparison of ML models. These allow scene understanding across several tasks, including classification, semantic/instance/panoptic segmentation, and change detection, etc. In the following subsections, we summarize related works to railway change detection, semantic segmentation, and we investigate the existing application in railway contexts and present the available datasets.

4.2.1 Railway change detection

Change in railway environments may affect safety and service continuity. Vegetation is a frequent cause of obstruction and can enter clearance zones near tracks or catenaries. For this reason, many studies aim to detect such changes early and in a reliable way. Vegetation monitoring along railway corridors is a critical operational task due to its direct implications on safety, signal visibility, and infrastructure integrity. Overgrowth or falling trees can obstruct tracks and overhead lines, causing delays or accidents. In response, infrastructure managers such as Network Rail have adopted frequent LiDAR-based surveys to enable early detection of vegetation encroachment [243]. Mobile laser scanning (MLS) systems have become the primary tool for capturing the geometry of the trackside environment at high resolution. These bi-temporal point clouds serve as the foundation for vegetation change detection. Early approaches, such as those introduced by Sturari et al. (2017), focused on robotic platforms combining LiDAR and imagery to identify structural anomalies [244]. However, these methods often lacked specificity regarding the type of change or its relevance to vegetation. More recent developments emphasize object-level analysis. Hirt et al. (2021) proposed a tree-centric strategy using occupancy grids to quantify structural evolution, enabling fine-scale tracking of height and volume changes [245]. This object-based formulation can help with the detection of targeted risks, such as individual trees entering the clearance envelope.

As we demonstrated in chapters 2 and 3, the use of semantic segmentation adds clear value. It helps isolate vegetation from other classes and avoids false alerts due to vehicles or pedestrians. It also improves the clarity of change results, which supports better planning and risk control. In short, existing work shows that 3D point clouds support change detection in railways, and that semantic segmentation can improve the focus and quality of the results, especially for vegetation.

4.2.2 Semantic segmentation for railways

Even if semantic segmentation has gained progress, particularly in urban environment, applications in the railway context remain limited. Arastounia [246] developed an automated method to recognize key components of railroad infrastructure from 3D LiDAR data. The used dataset covers a 550 m section of Austrian rural railroad and contains thirty-one keys element with only XYZ coordinate information. The method segmented rail tracks, contact cables, catenary cables, return current cables, masts, and cantilevers based on their physical characteristics and spatial relationships. Recognition involved analyzing local neighborhoods, objects shape, topological relationships, and employing algorithms like KD tree's structure [247] and FLANN's nearest neighbor search [248] for computational efficiency. The approach achieved 96.4% accuracy and an average of 97.1% precision at point level. Despite the high metrics, this methodology cannot be generalized, the parameters need to be adjusted, and it remains particularly sensitive to occlusion. Additionally, the data used is not publicly available.

Heuristic methods have a long history, but learning-based approaches are becoming more prevalent. The latter aims to lessen reliance on parameters and enhance generalization capabilities. Chen et al. [249] proposed a method for automatically classifying railway electrification assets using mobile laser scanning data. They used a

multi-scale Hierarchical Conditional Random Field (HiCRF) model to capture spatial relationships, improving classification accuracy compared to local methods. The model achieved an overall accuracy of 99.67% for ten objects classes. The authors in [250] proposed a deep learning-based methodology for the semantic segmentation of railway infrastructure from 3D point clouds. PointNet++ and KPConv were evaluated in four diverse scenarios, exhibiting a remarkable ability to adapt to varying data quality and density conditions. Notably, it achieved a mean accuracy of 90% when assessed on a 90 km long railway dataset used for both training and testing. They also evaluated the pros and cons of their approach, identifying the impact of intensity values as input features on segmentation results. Their best-performing tested architecture achieves a mean Intersection over Union (mIoU) of 74.89%. Furthermore, the methodology demonstrated its versatility by maintaining robust performance across different assets, such as rails, cables, and traffic lights, with an F1-score above 90% in all cases. Bram et al. [251] used PointNet++, SuperPoint Graph, and Point Transformer as three main deep learning models to perform semantic segmentation of railways catenary arches. The models were trained and assessed on the Railways Catenary Arches dataset with 14 labelled classes. PointNet++ achieved the best performance with over 71% in Intersection over Union (IoU).

The common disadvantage of these studies is that they do not evaluate their methods on the same dataset, in the same context, or with the same number of classes. This makes it almost impossible to make a fair comparison between all these studies, which in general, do not make their data publicly available.

4.2.3 Existing railways dataset

In the context of railway scene understanding, several image datasets have been developed. RailSem19 [252] offers 8500 annotated sequences from a train's perspective, including crossings and tram scenes. FRSign (French Railway Signaling) [253] contains over one hundred thousand images annotated over six French railway traffic lights with acquisition details (time, date, sensors information, and bounding boxes). RAWPED [254], [255] is dedicated to pedestrian detection in railway scenes. GERALD [256] features five thousand images and annotations for German railway signals. OSDaR23 [257], a multi-sensor dataset, captures various railway scenes with an array of sensors. These datasets are vital for developing and testing algorithms for railway scene analysis, promoting safety and automation in the industry. Compared to image datasets, there is a shortage of open 3D point cloud datasets. To date, we found the following:

- WHU-Railway3D: Introduced by Wuhan University of Science and Technology in 2023, this is a new railway point cloud dataset. It covers 30 km, contains four billion points, and is annotated using 11 classes. Its great advantage is that it covers three different environments, including urban, rural, and plateau. Each environment covers approximately 10 km: the urban railway dataset spans 10.7 km, captured using Optech's Lynx Mobile Mapper System in central China; the rural railway dataset, covering approximately 10.6 km, was collected through an MLS system with dual HiScan-Z LiDAR sensors; and, lastly, the plateau railway dataset, spanning around 10.4 km, was obtained using a Rail Mobile Measurement System equipped with a 32-line LiDAR sensor [258].

- Catenary Arch Dataset: Covering an 800 m stretch of railway track near Delft, Netherlands, this dataset captures fifteen catenary arches in high detail [259]. The data collection process involved the use of a Trimble TX8 Terrestrial Laser Scanner (TLS). This dataset provides the XYZ coordinates of points but does not include color information, intensity, or normal for the points. It is referenced within the Rijksdriehoeksstelsel coordinate system using meters as units. The number of points per arch varies from 1.6 million to 11 million, and each point is manually labelled into 14 distinct classes [259].
- OSDaR23: The Open Sensor Data for Rail 2023 dataset offers a comprehensive multi-sensor perspective of the railway environment, recorded in Hamburg, Germany during September 2021 [257], [260]. It uses a railway vehicle equipped with an array of sensors, including lidar, high and low resolution RGB cameras, an InfraRed camera, and a radar. The data were acquired through a Velodyne HDL-64E lidar sensor operating at 10 Hz frequency. These lidar data are further enriched with annotations of various object classes (20), such as trains, rail tracks, catenary poles, signs, vegetation, buildings, persons, vehicles, etc.
- Others dataset: Several other datasets have been used for railway scene understanding tasks, but they are not publicly available or irrelevant. They include the Austrian rural railroad [246], UA_L-DoTT [261], Vigo dataset [236], [250], TrainSim [262], INFRABEL-5 [263], Railway SLAM Dataset [264], MUIF [265] by the Italian Railways Network Enterprise (RFI), and MOMIT [266].

Existing 3D point cloud datasets for railway scene understanding are often restricted in size, variety, and semantic class, limiting their practical applicability. Our Rail3D dataset addresses these limitations by providing a comprehensive collection of railway environments from Hungary, France, and Belgium. This multi-context approach includes diverse railway landscapes, with various railway types, track conditions, and densities. The table below summarizes the datasets and their key specifications.

Table 12. Statistics overview of the existing railway point cloud datasets.

Dataset	Year	Additional Fields	Points/Sequences	Classes	Country
Catenary Arch	2021	Intensity	55 million	14	Netherlands
WHU-Railway3D	2023	Intensity, Number Of Returns	4 billion	11	China
OSDaR23	2023	Intensity	45 sequences	20	Germany
Rail3D	2024	RGB, Intensity	288 million	09	France, Hungary, Belgium

4.3 Materials and Methods

This section details the process of labelling the dataset and the methods of benchmarks. We begin by describing the data specifications used for both: change detection and for semantic segmentation and then the classes used and how the annotation process was conducted. Then, we describe the three methods for semantic segmentation in railway environments and use the trained model to classify the airborne data for vegetation change. **Figure 43** illustrates the full workflow of this chapter.

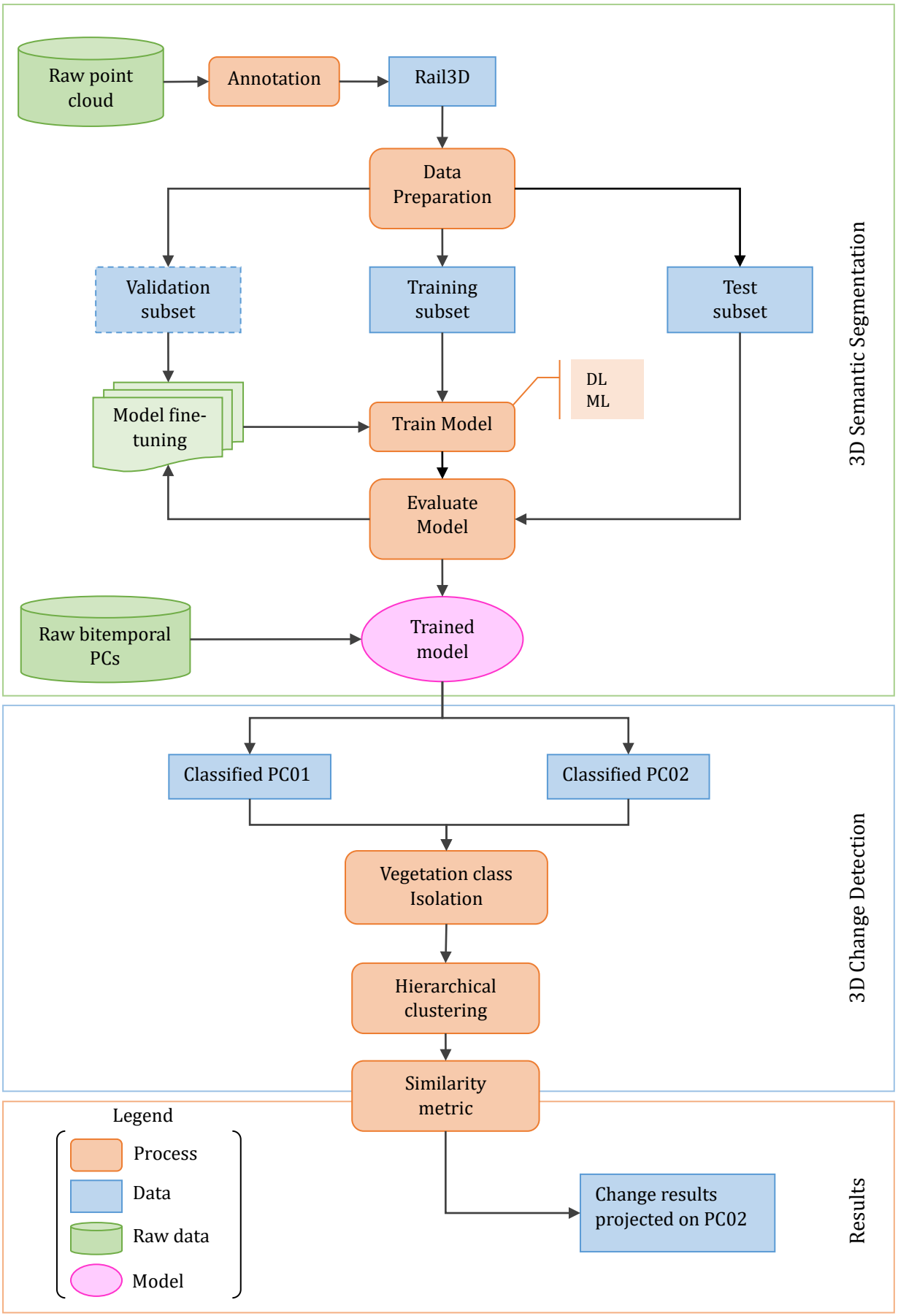


Figure 43. General workflow of chapter 04

4.3.1 Change detection data

The vegetation change detection analysis was conducted on a subset of airborne LiDAR data acquired over a railway section located to the west Apeldoorn, in the Netherlands (**Figure 44**). The data were made publicly available through the national geospatial platform spoorinbeeld.nl, which provides access to LiDAR datasets acquired for railway infrastructure analysis. Two epochs were selected for this study: the first, named heli_FP_201807_40972, corresponds to an acquisition carried out in July 2018 and contains approximately 18 million points; the second, heli_FP_20230628_41903-B22, was acquired in June 2023 and contains approximately 37 million points. Both datasets are delivered in LAS format and referenced in the Dutch national coordinate system, EPSG:28992 (Amersfoort / RD New). Based on the point distribution, acquisition patterns, and metadata, the data were assumed to have been collected using airborne LiDAR sensors mounted on a helicopter platform. The relatively dense and uniform point distribution, along with the extended linear coverage of railway corridors, indicates that the acquisition was likely performed for vegetation growth monitoring purposes.

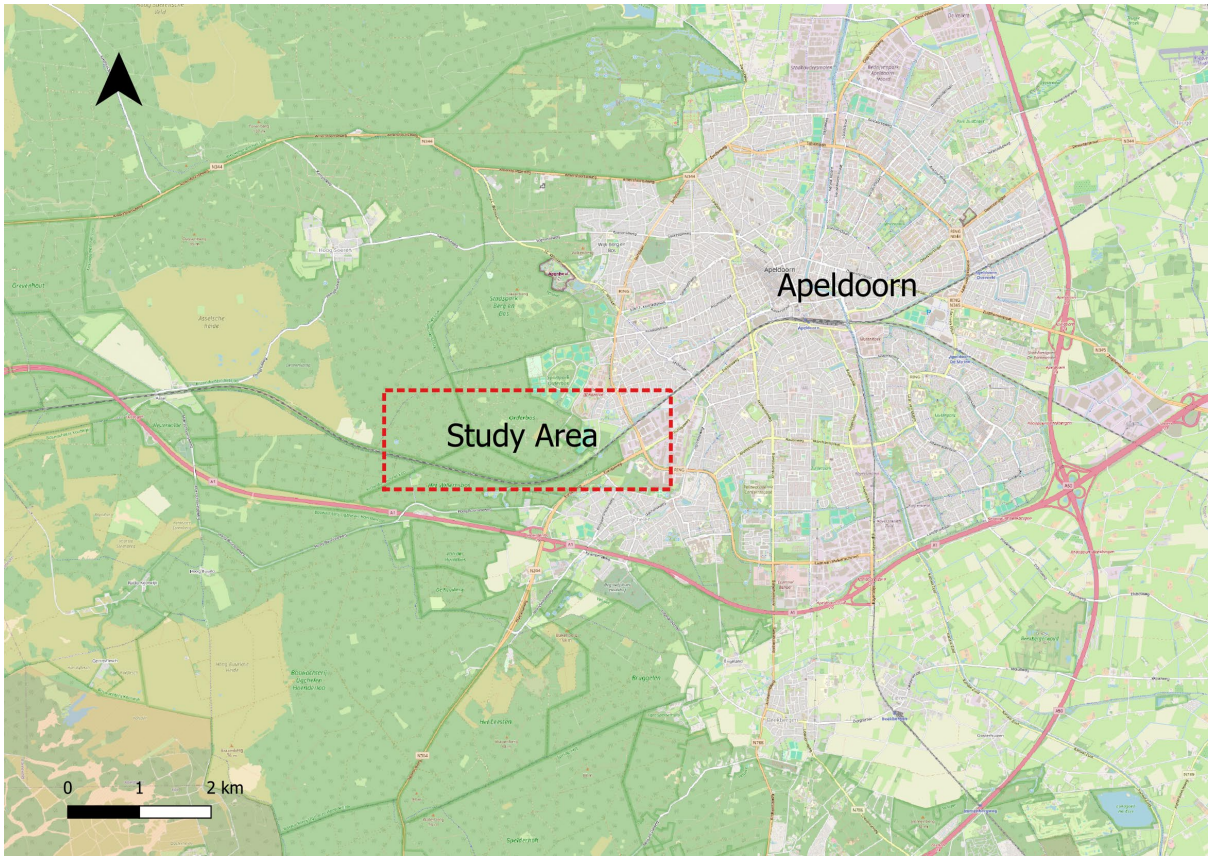
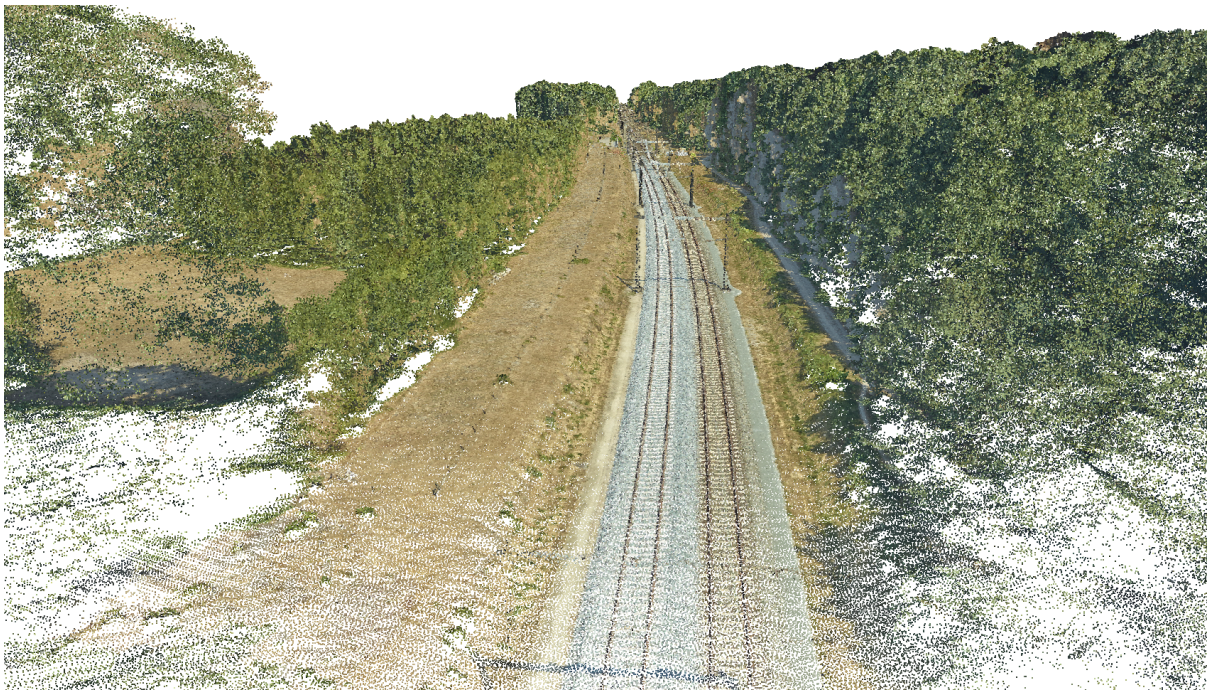


Figure 44. Location of the study area west of Apeldoorn, in Netherlands, marked with a red rectangle. © OpenStreetMap

The selected section spans approximately one kilometer along a railway corridor characterized by dense and heterogeneous vegetation. This segment was chosen due to the presence of trees and shrubs near the tracks, offering a representative use case for assessing vegetation growth and clearance. **Figure 45** provides a visual comparison of

the point clouds from both epochs, illustrating the evolution of vegetation structure over the five-year interval.



(a).

(b).

Figure 45. Visualization of the LiDAR point clouds from 2018 (**a**) and 2023 (**b**) showing changes in vegetation along the railway corridor.

4.3.2 Dataset specification

The Rail3D dataset is the first multi-context point cloud dataset for railways semantic understanding. It is composed of three distinct datasets from Hungary, France, and

Belgium, providing a diverse representation of railways. For this, we annotated three datasets named HMLS, SNCF, and INFRABEL. The firsts two datasets are publicly available, while the third one was provided by Infrabel, the Belgian railway infrastructure manager, under a confidentiality agreement. The datasets were acquired using different LiDAR sensors, ensuring a wide range of point densities and acquisition conditions. This diversity is crucial for developing robust and generalizable models for railway scene understanding. The specifications of each dataset are detailed in the subsections below.

Hungarian MLS Point Clouds of Railroad Environment

This LiDAR dataset represents a collection acquired by the Hungarian State Railways using the mobile mapping system (MMS)—Riegl VMX-450 high-density. These LiDAR scans were conducted from a railroad vehicle recording data at a rate of 1.1 million points per second. It exhibited high precision, with an average 3D range accuracy of 3 mm and a maximum threshold of 7 mm, ensuring high-quality data capture. The positional accuracy of the acquired point clouds averaged 3 cm, with a maximum threshold of 5 cm. This dataset not only contains georeferenced spatial information in the form of 3D coordinates but also incorporates intensity and RGB data, enhancing its utility for diverse applications. The dataset adheres to the Hungarian national spatial reference system, designated as EPSG:23700. Originally, three distinct datasets were thoughtfully selected, each representing different topographical regions within Hungary. The original data can be found in [267].

French MLS Point Clouds of Railroad Environment

The SNCF LiDAR dataset is a valuable resource provided by SNCF Reseau, the French state-owned railway company. This dataset comprises approximately 2 km of rail-borne LiDAR data, representing a non-annotated subset of a more extensive collection. The geospatial reference system employed for this dataset is RGF93-CC44/NGF-IGN69, ensuring its alignment with regional mapping standards and geographic referencing. The dataset is primarily presented in compressed LAZ format with a total of 16 tiles publicly available under the Open Database License (ODbL).

Belgian MLS Point Clouds of Railroad Environment

For several years, Infrabel, the Belgian railway infrastructure manager, was using LiDAR technology to acquire data over the country's railways network. The lidar acquisition process involves mounting a Z + F 9012 lidar sensor on the front part of a train (EM202 vehicle) and recording the point cloud data as the train travels along the tracks. The train collects data for every railway line in Belgium at least twice a year. This is highly relevant for 3D change detection, which we will be looking at in our next studies. In addition to LiDAR, four cameras are used to capture the colors, two at the front and two at the back. However, for this application, we only used the point cloud without intensity and without colors. The point clouds were encoded in LAS format and the coordinates were in Lambert Belge 72. We chose three different areas in Belgium: Brussels, mid-way between Brussels and Ghent, and the south of Ghent. Each of these contexts presented a different challenge, either in terms of occlusion, complexity, or unbalanced classes.

New addition: Dutch MLS Point Clouds of Railroad

To complement the original Rail3D dataset and extend its representativeness, two new annotated point cloud sections were added, both located in the Netherlands (**Figure 46**). The first corresponds to a double-track segment in the urban center of Culemborg, spanning approximately 1160 meters. The second is a single-track section located south of Veenendaal, covering 1400 meters. These scenes were selected for their contrasting geometries and surrounding contexts, thereby supporting the construction of models capable of generalizing across different railway environments.

The raw LiDAR data were acquired from the national geospatial platform https://spoorinbeeld.nl, using mobile sensor and referenced in the EPSG:28992 coordinate system. Annotation was initially conducted by Milo Beliën as part of his Master's thesis on point cloud segmentation and subsequently refined by the authors to ensure consistency with the existing Rail3D taxonomy. In addition to the original semantic classes, a new class (label 10) was introduced to capture dynamic or transient objects, including trains, pedestrians, and infrastructure elements that are not persistently present. This extension facilitates a more inclusive representation of railway scenes, particularly in urban contexts where such elements are common.

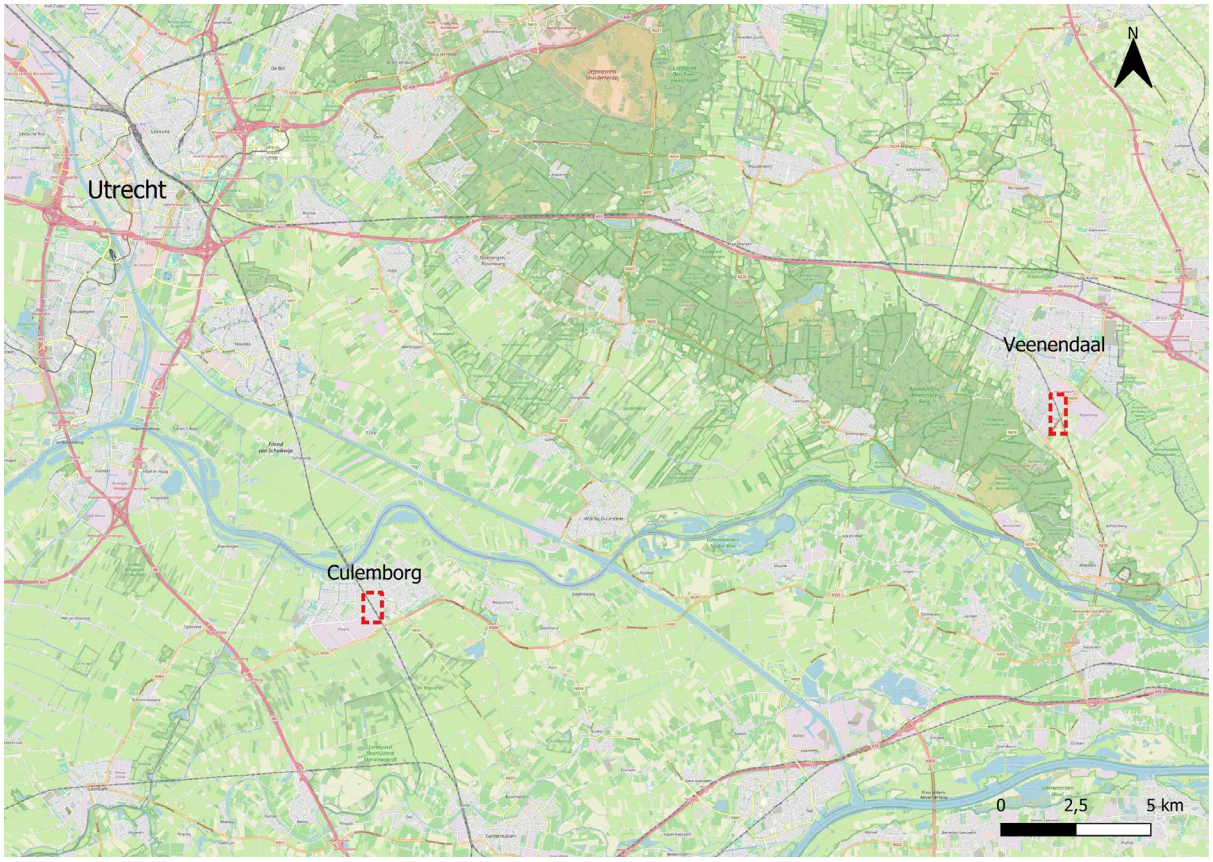
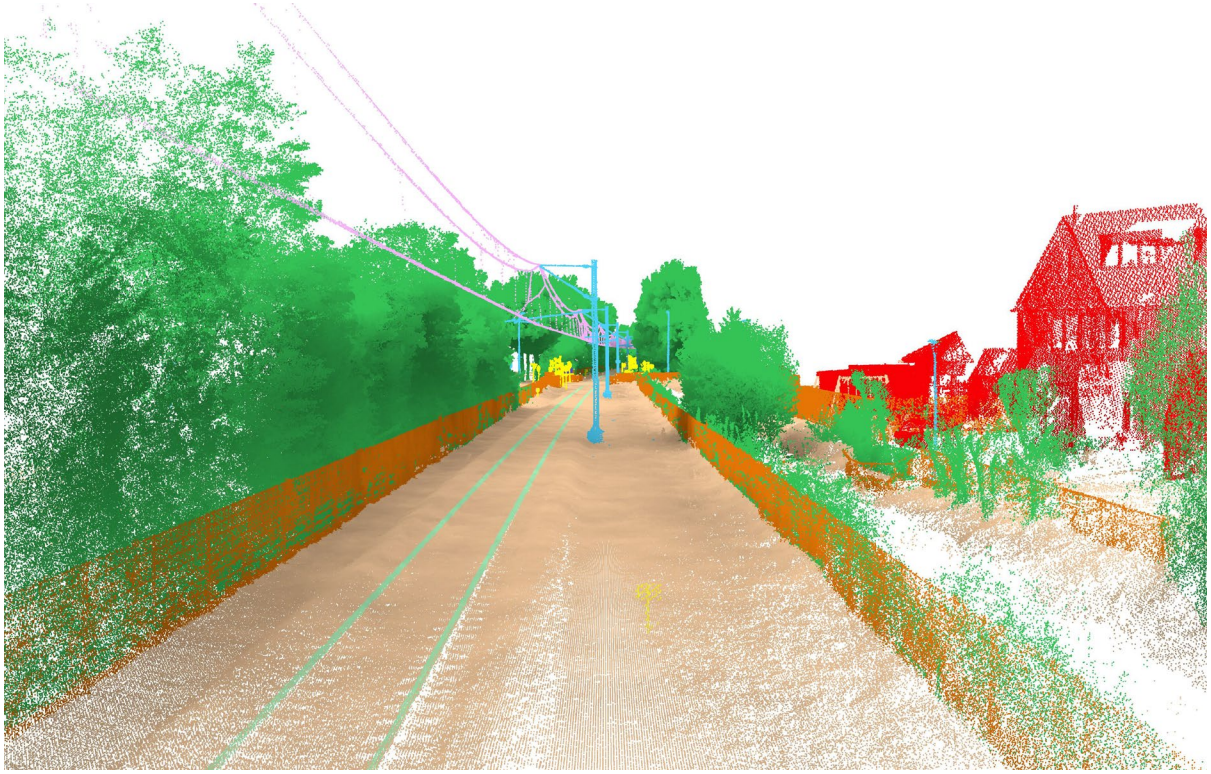


Figure 46. Location of the double track section in the center of Culemborg, and the single track south of Veenendaal. Sections marked with a red rectangle. © OpenStreetMap

The integration of these new scenes is aligned with ongoing developments in the field of railway semantic segmentation, as evidenced by several recent large-scale datasets. The RailPC dataset [268] emphasizes the relevance of detailed semantic hierarchies,

incorporating 16 classes derived from Chinese railway infrastructure. The dataset demonstrates that increased class granularity contributes to more precise segmentation and improved generalization, particularly in complex environments. However, its geographic scope remains limited, highlighting the need for complementary datasets in other regional contexts. Similarly, The RailCloud-HdF dataset [269] contributes a long-range acquisition campaign over 267 km of French railway. It includes 9 classes and focuses on balancing label distribution to mitigate class imbalance during training. The results obtained using several established segmentation architectures highlight the challenges associated with rare object classes and dynamic scene elements, reinforcing the importance of dataset diversity and quality annotations.

Taken together, these contributions underline three recurring challenges in railway point cloud segmentation: (i) the limited variety of track types and scene configurations within individual datasets, (ii) the frequent exclusion of non-static or transient classes, and (iii) the underrepresentation of European railway infrastructure. The Dutch extensions to Rail3D address these limitations by adding annotated data that differs both structurally and contextually from the initial dataset, while remaining consistent in terms of acquisition method and annotation scheme as can be seen in figure below.



(a).

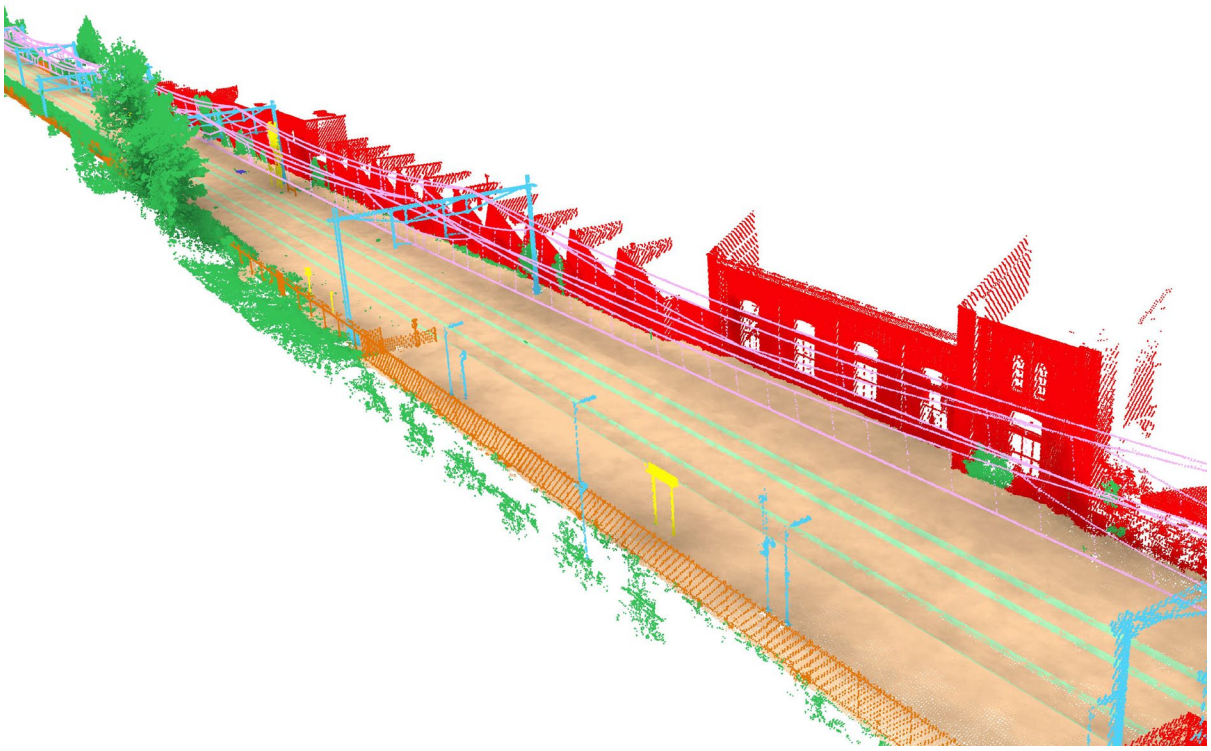


Figure 47. Sample views from the annotated Dutch railway section. Where (a) is the single track, and (b) represent a perspective view of the double track

As a summary, we provide, in **Table 13**, an overview of the Rail3D datasets, detailing key metrics such as total points, total length, attributes, and the number of tiles per set (for train, validation, and test). Overall, for each railway scene, the training data accounted 60%, the validation data accounted for about 20%, and the test data ac-counted for about 20%.

Table 13. Description of datasets in numbers of points, length (in meter), attributes, and number of tiles per set (train, validation, and test).

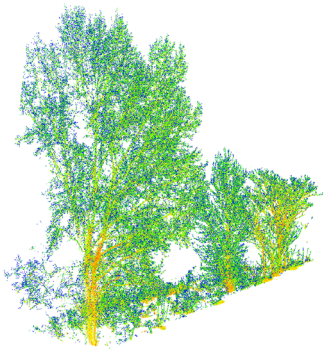
Dataset	Total Points	Total Length	Attributes	Train	Validation	Test
HMLS	104,958,796	2140 m	X, Y, Z, R, G, B, Intensity	17	6	6
SNCF	143,313,588	1600 m	X, Y, Z, R, G, B, Intensity	10	3	3
INFRABEL	39,648,006	2087 m	X, Y, Z, Intensity	12	4	5
Dutch	-	2500 m	X, Y, Z, Intensity	-	-	-

4.3.3 Classes typology

To choose class labels, we combined literature reviews [250] with insights from railway industry experts from Infrabel. This approach helped us to identify classes critical for railway infrastructure applications, ensuring our labels reflect both academic and practical considerations. This resulted in the following list of classes and their corresponding objects as illustrated in **Figure 48**:

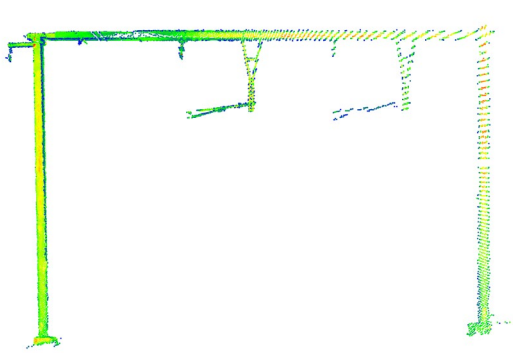
- Ground (label 1): represents various ground surfaces, including rough ground, cement, and asphalt.
- Vegetation (label 2): covers all types of vegetation present in the railway environment, such as trees and median vegetation.
- Rail (label 3): refers to the physical track structure.
- Poles (label 4): includes all types of poles found along the railway, such as catenary poles and utility poles.
- Wires (label 5): covers various overhead wires, including catenary wires, power lines, and communication cables.
- Signaling (label 6): represents all types of railways signaling equipment, such as traffic lights and traffic signs.
- Fence (label 7): includes several types of fences and barriers used along the railways, such as security fences, noise insulation panels, and guardrails.
- Installation (label 8): refers to larger structures that are part of the railway infrastructure, such as boxes and passenger cabins.
- Building (label 9): covers all types of buildings located within the railway corridor, such as houses, warehouses, and stations.
- Moving objects (label 10): capture dynamic or transient objects, including trains, pedestrians, and temporal infrastructure elements.

Ground

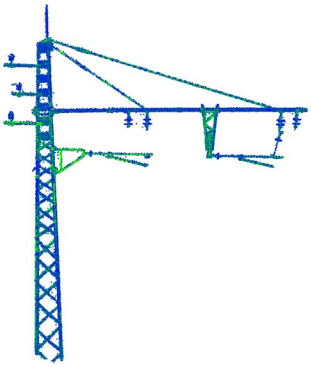


Vegetation

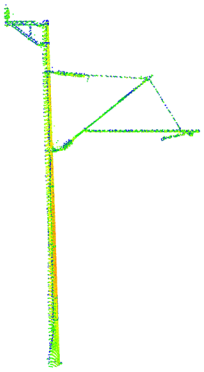
Rail



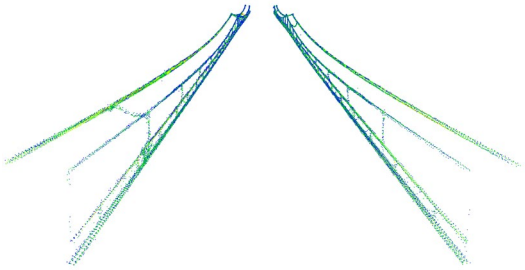
Pole (INFRABEL)



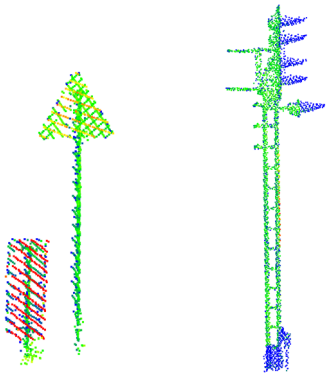
Pole (SNCF)



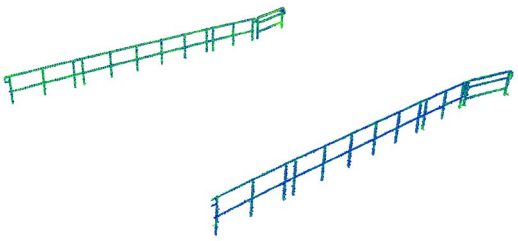
Pole (HMLS)



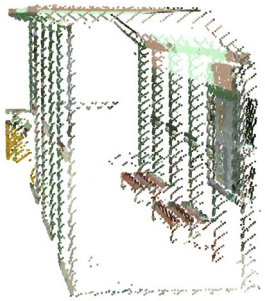
Wires



Signaling (HMLS)



Fence



Installation

Building

Figure 48. Illustration of each object class from the three datasets.

4.3.4 Data annotation

The quality of the annotation is a determining factor for a good dataset, as well as for all the successive tasks. As shown by Silvia et al. (2023)'s results in [270], the annotation differs from one person to another, even for the same scene and with the same class definition. This has a significant impact on the results of deep learning semantic segmentation models, with a relationship of R2 = 0.765 between the points labelled in concordance and the F1-score. Even if the discordances do not exceed 1% in general, we tried to perform an entirely manual annotation, without using any automatic algorithms. The annotation was performed on CloudCompare [271] (version 2.13.0) using the QCloudLayers tool and was controlled by a second and a third annotator. They

were given guidelines to help them with the annotation process, but it was up to them to make the final decision on the class label for each point. **Table 14** details points per class, revealing dominant vegetation and ground classes. The minority classes, like signals and buildings, may pose challenges for the semantic segmentation models.

Table 14. The Class distribution of each subset.

Dataset	Ground	Vegetation	Rail	Poles	Wires	Signaling	Fences	Installation	Building
HMLS	69,712,961	32,208,451	1,157,524	667,108	430,949	74,687	359,907	74,272	272,937
SNCF	51,214,064	85,506,927	2,592,131	1,390,768	2,187,299	110,467	224,576	87,356	0
INFRABEL	33,698,796	1,381,599	2,041,258	654,757	1,735,242	24,926	51,100	60,328	0

In **Figures 49–51**, we show samples of our annotation. Each color code represents a class, as indicated in the legend.

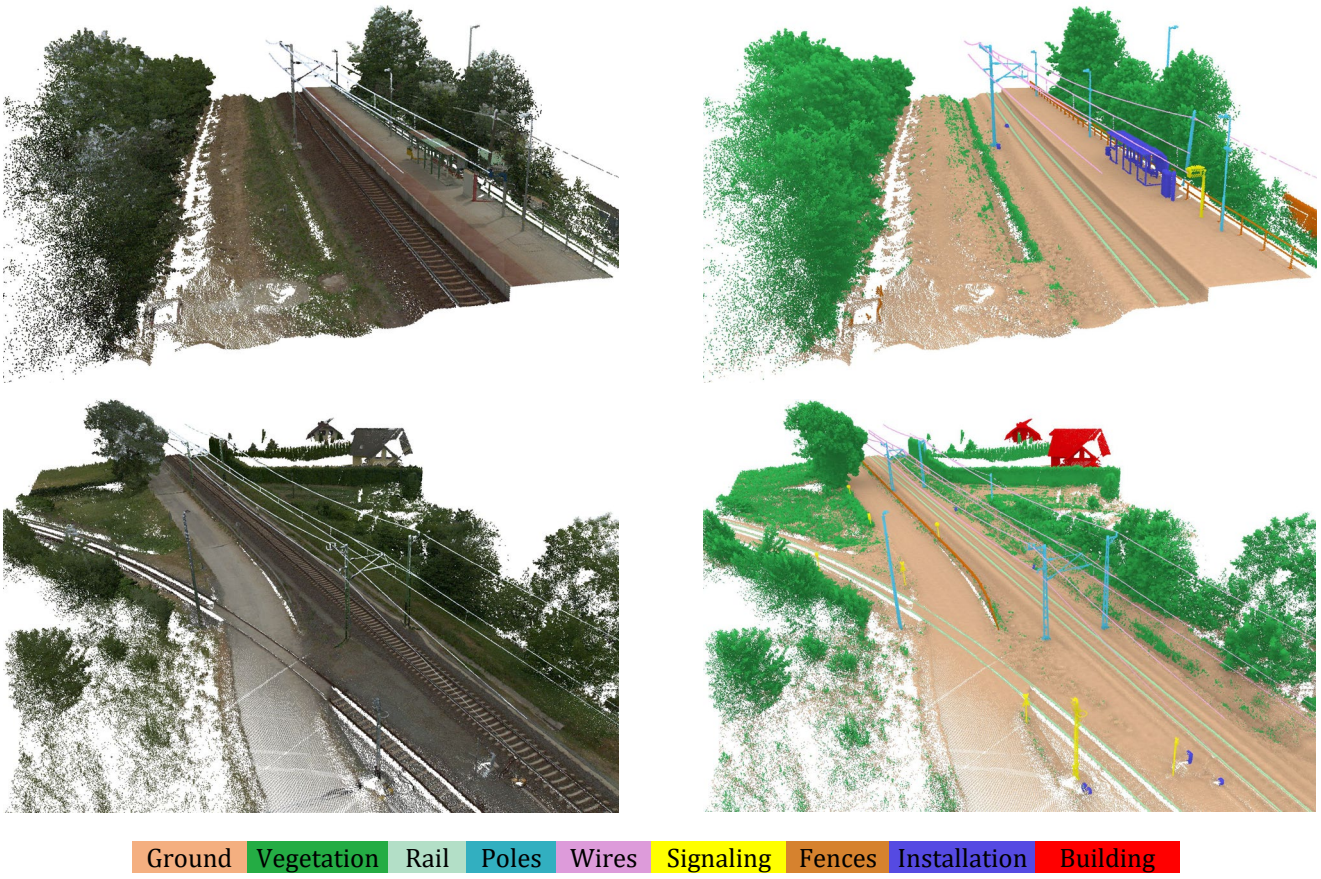


Figure 49. Examples of our annotation of the Hungarian MLS dataset, displayed in colors (left) and corresponding classes (right). The legend is valid for the rest of Figures.

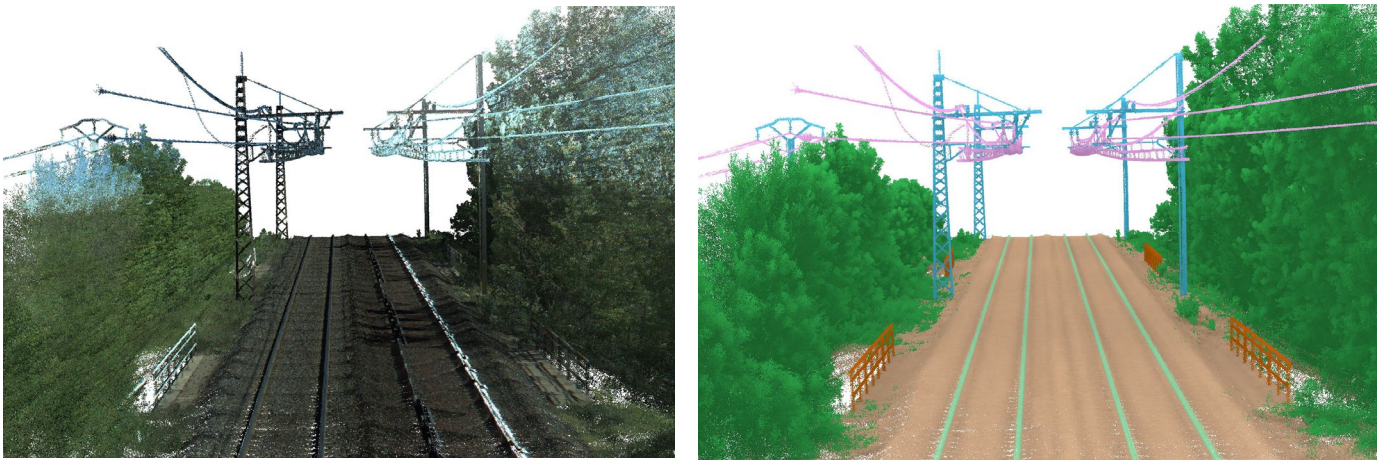


Figure 50. Examples of our annotation of the SNCF dataset, displayed in colors (left) and corresponding classes (right).

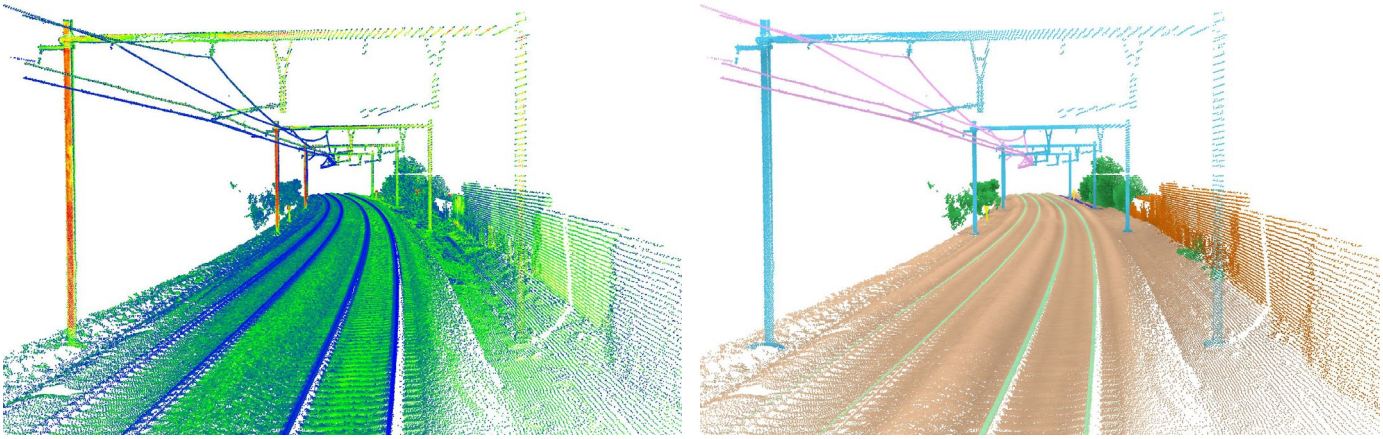


Figure 51. Point cloud from the INFRABEL dataset, displayed in intensity (left) and corresponding labels (right).

4.3.5 Semantic segmentation methods

In this subsection, we explore three baseline models for semantic segmentation of our Rail3D dataset: KPConv, LightGBM, and 3DMASC. KPConv, known for its novel convolution operation tailored for point clouds, stands out for its efficiency in semantic segmentation tasks and it shows high performance over several datasets [238], [240]. LightGBM, a gradient-boosting framework, is chosen for its accuracy and high efficiency, particularly when dealing with handcrafted features from point clouds [272]. 3DMASC, the most recent one, offers an accessible and explainable approach for point cloud classification, managing diverse datasets effectively [273]. While KPConv has shown exceptional performance in various benchmarks, LightGBM and 3DMASC are well-established in the literature, offering robust solutions for point cloud classification. Our choice of these models is based on their proven capabilities and the recent advancements they represent in the field.

Deep Learning Baseline

To evaluate semantic segmentation models using our datasets, we investigated the Kernel Point Convolution (KPConv) [145]. KPConv offers a novel convolution operation specifically designed for point clouds. It departs from traditional grid-based convolutions by utilizing kernel points to define local support regions. These kernel points possess learnable weights and capture the spatial relationships between points within their area of influence. Deformable KPConv further enhances this capability by adapting the location of kernel points according to the point cloud's geometry (**Figure 52**) We opted for KPConv due to its ability to efficiently handle point cloud data and its demonstrated success in semantic segmentation tasks [274].

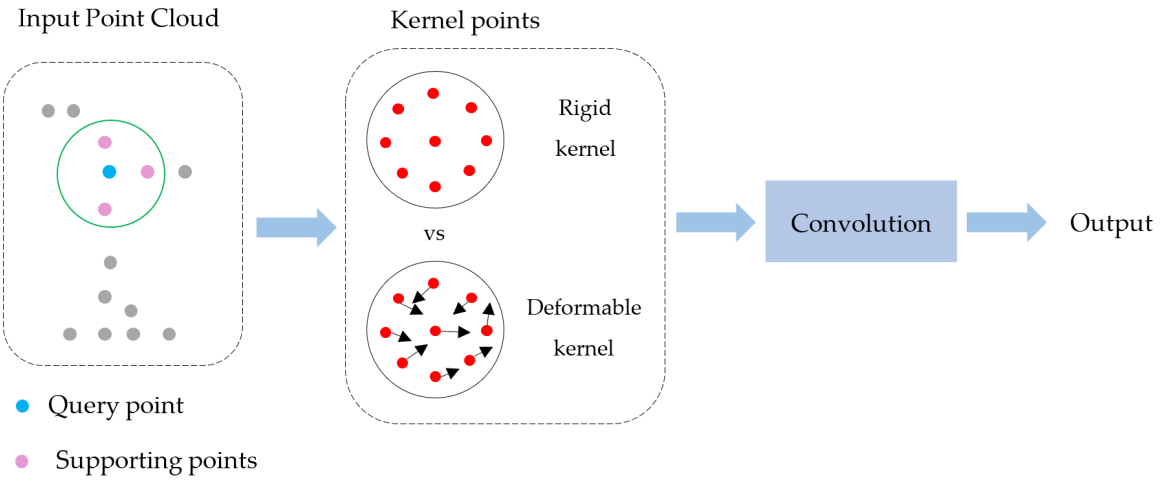


Figure 52. KPConv defines a set of kernel points to aggregate local features and performs convolution on supporting points, which are selected from a range search around the query point. (Illustration adapted from [275]).

The KP-FCNN model is built as a fully convolutional network designed for point cloud semantic segmentation. Its encoder has five layers; each layer has two convolutional blocks. The first block in each layer reduces the size of the input, except for the very first layer. These blocks are like ResNet blocks but use KPConv for the convolution process, along with batch normalization and leaky ReLU activation. After the encoding process, global average pooling collects the features, which are then passed through fully connected and SoftMax layers, like how traditional image CNNs work. The de-coder part of the model uses the nearest neighbor upsampling to improve the detail of point-wise features, using skip connections to combine features from both the encoder and decoder at different points. The KP-FCNN architecture is detailed in the bellow illustration in **Figure 53**.

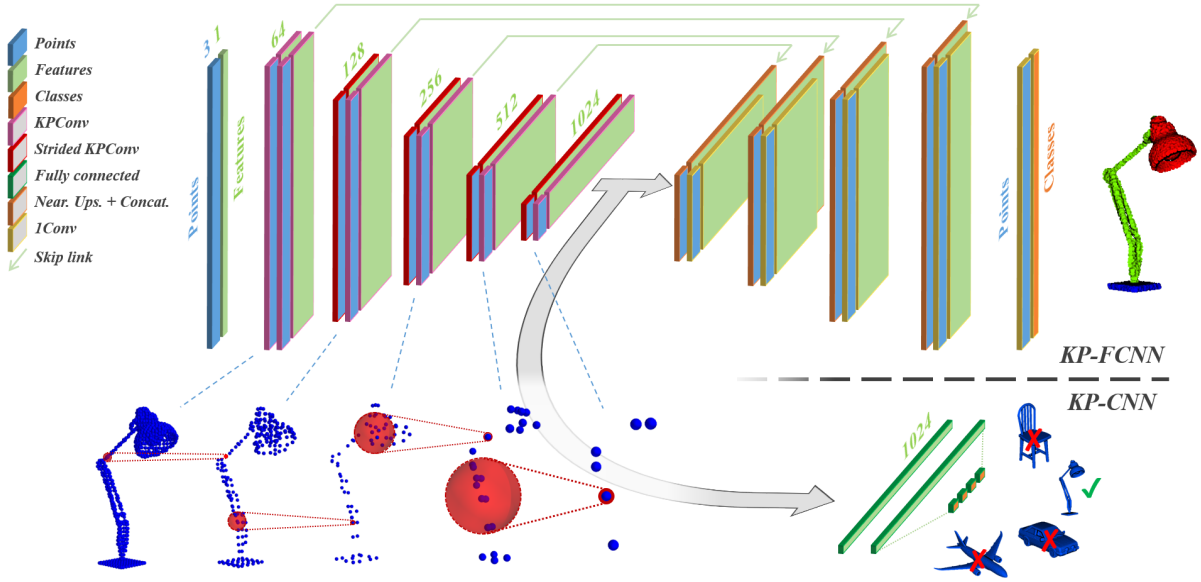


Figure 53. Illustration of the Kernel Point Convolution network architecture [145] used as a baseline method in our study. The top illustration is for semantic segmentation; the bottom one is for the classification task.

To handle the large size of point clouds, the model works with smaller portions of the point cloud, shaped into spheres with a set radius. This method helps manage the large amount of data effectively. Our use of the model follows the original training guidelines, including how big the input spheres are and how the kernel points are arranged. However, we simplified the input features to just a constant value (equal to 1) that represents the shape of the data points. This simplification, along with using a sphere-based method for testing and a voting system to decide the final segmentation, helps keep the segmentation accurate even with the large size and complexity of rail-way point clouds.

Machine Learning Baseline

The extraction of meaningful information from raw 3D point cloud data is crucial for the effective application of machine learning algorithms in classification tasks. We opted into the use of handcrafted features, which are carefully selected descriptors that encapsulate the geometric, spectral, and local dimensional characteristics of the point cloud. The process of feature selection and extraction is important for enhancing the capability of machine learning models to conduct accurate classifications. By selecting the right handcrafted features, the classification algorithms (Random Forest and LightGBM) can achieve a more robust understanding and interpretation of the complex railway assets. We followed the general steps:

1. Feature identification: this selection process aims to identify a set of informative attributes that effectively capture the characteristics of the points and their surrounding context. These attributes can be derived from various sources including the point cloud's geometry, spectral information, and local dimensionality. Common geometrical features include eigenvalues of the neighborhood covariance matrix which help identify local shapes like lines, planes, contours, edge, and volume. Additionally, point density and verticality estimations can be informative for

classification. In the case of lidar data, features based on multiple returns, such as the number of returns or return ratio, are valuable for distinguishing ground, buildings, and vegetation. Height variations (e.g., Z range) and radiometric information like backscattered intensity further enhance classification, especially when dealing with objects that share similar geometries.

2. Feature extraction: it involves analyzing a point's local neighborhood to capture its geometric properties. The choice of neighborhood type, whether spherical, cylindrical, or cubic, along with the decision to compute descriptive features at single or multiple scales significantly influences the classification outcome. Multiscale analysis has shown greater descriptive power, capturing scene elements of varying sizes and the geometric variations of objects more effectively. While optimal parameterization, including neighborhood type, number of scales, and scale radius, is crucial, automatic scale selection based on minimizing redundancy and maximizing classification accuracy is an active area of research.
3. Feature selection: high-dimensional data can harbor irrelevant or redundant features that inflate model complexity and hinder learning. Selection methods aim for an optimal feature subset, maximizing informativeness while minimizing redundancy. Filter-based methods rank features using relevance scores (e.g., Fisher's index or information gain index) but may miss feature synergies. Wrapper and embedded methods integrate the classifier, using accuracy (wrapper) or feature importance (embedded) for selection [273].
4. Classification: this involves assigning labels to individual points within a 3D point cloud. Common approaches include instance-based methods like k-nearest neighbors (kNN) and Support Vector Machines (SVMs), as well as ensemble methods like Random Forests (RFs) and gradient-boosting algorithms like LightGBM. RFs remain popular due to their ease of use, robustness to overfitting, and ability to provide feature importance metrics. However, these individual point classifiers, including RFs, struggle to incorporate spatial context beyond what is captured in the feature vector. Contextual classification methods address this by explicitly modelling relationships between neighboring points during training. While techniques like Conditional Random Fields (CRFs) can achieve higher accuracy by leveraging spatial context, they are computationally expensive and may not perfectly capture complex relationships due to training data limitations. Our tests leverage the strengths of LightGBM, a powerful gradient-boosting algorithm, and Random Forest, while indirectly incorporating contextual knowledge through carefully chosen feature descriptors.

Our first chosen 3D point cloud classification approach relies on handcrafted features and the LightGBM Classifier, a highly efficient and accurate gradient-boosting framework. We have carefully outlined a set of steps to improve classification performance (**Figure 54**). Given an annotated point clouds set, we extracted features, focusing on attributes like spatial distribution and local context. We used established features based on eigenvalues, as summarized in **Table 15**. To manage the unordered and varying density of 3D point clouds, we utilize a multi-scale and efficient spherical neighborhood for feature computation. Then, a concatenation step of features was performed before training the classifier. Based on the features' importance results and

the performance metrics, we can even use the model directly or once it has been retrained again with only the most important features.

Table 15. Used handcrafted features [273], each one calculated at a multi-scale sphere diameter: 0.5 m, 1 m, 1.5 m, and 2 m.

Features Category	Acronym	Description
Neighborhood features	PCA1	PCA: λ 1/(λ 1 + λ 2 + λ 3)
	PCA2	PCA: λ 2/(λ 1 + λ 2 + λ 3)
	PCA3	PCA: λ 3/(λ 1 + λ 2 + λ 3)
	SPHERICITY	(λ 3/λ 1)
	LINEARITY	(λ 1–λ 2)/λ 1
	PLANARITY	(λ 2–λ 3)/λ 1
	VERTICALITY	$\left\lvert\frac{\pi}{2} - angle(e_i, e_z)\right\rvert_{i\epsilon(0,2)}$
	ZRANGE	Zmax—Zmin
	ZMAX	Zmax—z
	ZMIN	Z—Zmin
	ANISOTROPY	Anisotropy measure
	DIP	Computed from PCA
	NORMDIP	Dip Angle (point cloud must have Normal)

The eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \in \mathbb{R}$ and the corresponding eigenvectors $e_1, e_2, e_3 \in \mathbb{R}^3$ of the neighborhood covariance matrix, defined by equation below. Where p is a given point and $\bar{p}$ is the centroid of the neighborhood N (the sphere points) [276]:

$$Cov\ (N) = \frac{1}{|N|}\sum_{p\in N}(p - \bar{p})(p - \bar{p})^T. \qquad (13)$$

The DIP feature extracted from principal components analysis (PCA) refers to the orientation or inclination of the surface at a point in the point cloud, relative to a vertical reference. Specifically, the DIP measures how much a surface "dips" away from the horizontal plane. NORMDIP (Normal Dip Angle) refines the concept of DIP by considering the angle of the surface's inclination (or dip) relative to the normal (perpendicular) vector of the point cloud's surface at that point. This feature requires that the point cloud has Normal calculated for each point, which represent the perpendicular direction to the surface at each point.

The use of multi-scales is essential for capturing the variability within railway scene point clouds. This method, in contrast with single-scale approaches, enhances

classification performance by adapting to the diverse sizes and shapes present in the data. The selection of scales was systematically parameterized to account for object diversity and neighborhood configurations, optimizing feature extraction across different scales. For feature extraction, we utilized eigenvalues from the neighborhood covariance matrix of each point to evaluate the point cloud geometry, focusing on attributes such as linearity, planarity, sphericity, etc. (see **Table 15**). Feature selection was conducted using embedded strategies, notably leveraging Random Forest-based metrics to identify the most informative features (feature importance). The classification phase employed LightGBM, chosen for its gradient-boosting capabilities and efficiency in processing the selected features. The parameters for LightGBM were determined through a grid search.

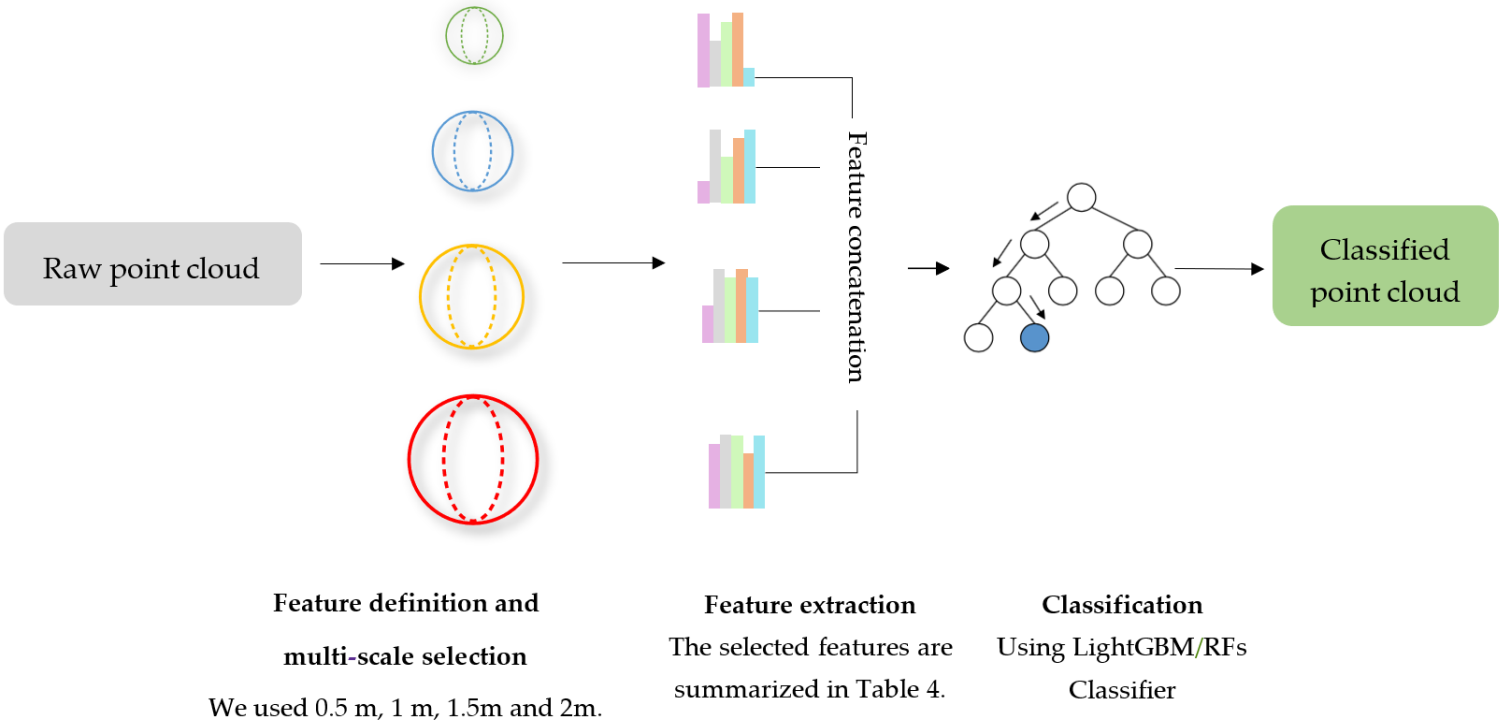


Figure 54. Multi-scale neighborhood selection, feature extraction, feature concatenation, and classification using LightGBM and Random Forest.

The second model is 3DMASC [273], which represents an accessible and explainable method for point cloud classification. It manages several attributes, scales, and point clouds, enabling the processing of diverse point cloud datasets, including those with spectral and multiple return attributes. The feature extraction process incorporates classical 3D data multi-scale descriptors, novel spatial variation features, and dual-cloud features, ensuring a comprehensive capture of both geometrical and spectral information. Subsequent steps in scale selection and feature fusion contribute to enhanced classification performance. The method is based on Random Forest as a classification model and CloudCompare for feature extraction, selection, importance, and classification. These are calculated either based on a spherical neighborhood or the k-nearest neighbors. **Table 15** showed a summary of the features we used for this test. In

Random Forest classification, feature importance is determined theoretically by evaluating how each feature contributes to reducing uncertainty or impurity in the model's predictions across all decision trees within the forest. This process involves calculating the decrease in impurity, often measured using the Gini impurity or entropy, attributed to each feature at every split point in each tree. The model aggregates these decreases across all trees to assign an importance score to each feature. The more a feature decreases impurity on average across all trees, the more important it is. This method allows us to identify which features play a crucial role in the model's decision-making process, highlighting their relative importance in predicting the target label.

4.3.6 Change detection method

Vegetation change detection was carried out using two airborne LiDAR datasets acquired over the same railway corridor in 2018 and 2023. The point clouds were first clipped to the common area of interest, ensuring spatial consistency in EPSG:28992. Each epoch was independently processed using the KPConv model trained on the Rail3D dataset. Despite domain differences, the trained network was able to generalize and produce reliable semantic predictions for the vegetation class.

The classified vegetation points from both epochs were then segmented using a hierarchical multi-scale approach based on the ℓ_0-cut pursuit algorithm. This method was inspired by the TreeIso framework, originally developed for individual tree segmentation from terrestrial LiDAR, but here adapted to airborne LiDAR data and linear railway environments. The segmentation process was structured into three levels, progressively aggregating points into larger spatial units. In the first level, fine-scale segmentation was performed to group points into small, homogeneous patches (referred to as superpoints) based on local geometry and spatial proximity. This stage used a low regularization parameter ($\lambda = 2$), which helps preserve detailed structures by limiting overmerging of nearby points. In the second level, these superpoints were iteratively merged into larger clusters based on spatial adjacency and geometric similarity, taking into account attributes such as elevation distribution and surface continuity. A moderate regularization value ($\lambda = 8$) was used to balance detail retention with structural coherence, enabling the emergence of mid-scale vegetation components, such as foliage masses or crown sections. The third level of the hierarchy focused on aggregating these mid-scale segments into complete vegetation objects by considering broader structural patterns and topological connectivity. Unlike the first two stages, this step did not apply a new regularization parameter; instead, it relied on geometric criteria such as bounding box overlap, topological adjacency, and connectivity in 3D space to group segments belonging to the same vegetation entity. This stage enables the abstraction of well-formed individual vegetation units such as shrubs or trees.

To detect vegetation changes between the two epochs, a robust object-level matching approach was employed. For each object in the recent epoch (2023), all candidate segments from the past epoch (2018) within a 1.5 meter radius were retrieved using a spatial KD-tree. When multiple candidates were found, the best match was selected based on feature similarity. Each segment was described by three geometric descriptors: centroid (3D mean position), the 95th height percentile (Z95), and the eigenvalues of the covariance matrix, capturing the object's spatial distribution. Similarity was

assessed using cosine similarity over the normalized feature vectors. A segment in 2023 was considered unchanged if at least one sufficiently similar object was found in 2018 (similarity score > 0.9) and changed otherwise. This method addresses the limitations of centroid-based distance alone, which can be misleading due to irregular shapes or density differences and provides a more robust characterization of vegetation evolution.

To evaluate the relevance of this object-based semantic approach, the results were compared to those obtained using two geometric baseline methods: cloud-to-cloud (C2C) and multi-scale model-to-model cloud comparison (M3C2). While they provide a scalar estimate of surface change, they do not produce structured outputs at the object level. In contrast, the proposed method integrates semantic context and delivers interpretable change directly aligned with operational needs for vegetation management.

4.4 Experiments and Results

This section evaluates the three chosen state-of-the-art chosen methods (KPConv, LightGBM, and 3DMASC). We also report on the application of the trained KPConv model to a separate airborne dataset for object-based vegetation change detection. The first subsection describes the implementation of each method to ensure replicability. The second subsection discusses the metrics used for evaluation as well as the results of the different tests performed in our study.

4.4.1 Implementation

These experiments required the creation of python code and the implementation of the original experiments. For 3DMASC, from feature extraction to inference, it was performed using the 3DMASC's CloudCompare Plugin and following the training procedure in [273]. Whereas, for KPConv, we utilize PyTorch original implementation. For the training experiment we used the same procedure as in [145]. For LightGBM, we used CloudCompare Plugin for feature extraction and feature importance. Then, we created Python scripts using the LightGBM framework (version 4.3.0) for classification. The code is available at https://github.com/akharroubi/Rail3D accessed on 14 December 2023. We adopted the training procedure in [79]. All these codes were accessed on 14 January 2024 and experiments were conducted on a workstation with a NVIDIA Ge-Force RTX 3090 graphics card and an i9-10980XE CPU @ 3.00 GHz with 256 GB RAM.

For the first experiment, we used KPConv without any changes to the network architecture [145]; only the hyperparameters were adapted. We used 15 kernel points with an input sphere radius of 2.5 m and a first subsampling grid size of 0.06 m to capture fine details. The convolution radius was set to 2.0 m, with a deformable convolution radius of 4.0 m for flexible feature learning. The kernel point influence was 'linear', and the aggregation mode was 'sum'. We initialized the feature's dimension at 128 and employed batch normalization with a momentum of 0.02. The model was trained for a maximum of 400 epochs, with a learning rate of 1 × 10−2, a batch size of 5, epoch steps of 500, and a validation size of 50, optimizing for both accuracy and efficiency. In contrast to the original article, however, we did not add any input features

(only X, Y, and Z coordinates), except for a constant feature equal to one, which encoded the geometry of the input points.

For the LightGBM experiment, we used the original training procedure as in [272]. When selecting a method for neighborhood extraction from point clouds, we considered cylindrical, nearest neighbor (k-NN), and spherical approaches. We chose the spherical neighborhood method. This decision was informed by research indicating that cylindrical neighborhoods can be more advantageous in extracting useful features from point clouds [276]. The spherical method was selected for its ability to uniformly capture the local structure around a point in all directions, providing a more balanced and comprehensive understanding of the point's immediate environment. This approach ensures that the features extracted are representative of the point's spatial context. For the parameters, we set the boosting_type to 'gbdt' for gradient-boosting decision tree, with num_leaves at 100 and max_depth at 30 to allow for complex model learning. The learning_rate was set to 0.05, with n_estimators at 100 for the number of boosting rounds. We also configured subsample_for_bin at 200, subsample at 0.8 for the subsample ratio of the training instance, and min_child_samples at 20 to ensure sufficient data for each tree. The class_weight was set to "balanced" to address class imbalance. We used the dataset split detailed in **Table 13** for training, validation, and testing across all models, ensuring consistency in model evaluation and comparison.

For the 3DMASC experiment, we used the Random Forest method with a spherical neighborhood for feature extraction and four scales (0.5 m, 1 m, 1.5 m, and 2 m). We set the max_depth of the trees to 30 and limited the max tree count to 150, allowing the model to learn detailed features without becoming overly complex. The min sample count was set to 10, ensuring that each leaf had enough samples to make reliable decisions. The 3DMASC plugin utilizes Random Forests to inherently assign importance scores to each of the 60 features (10 features at 4 scales) employed in the analysis. To enhance the interpretability of these scores, the plugin addresses redundancy through a two-step selection process. First, a bivariate pre-selection based on information gain (IG) and Pearson's correlation removes highly correlated features, resulting in a set with clearer importance scores. Second, the plugin implements an embedded back-ward selection using the RF feature importance. This process iteratively removes features with the lowest importance while monitoring the impact on model performance via Out-of-Bag (OOB) scores. The selection stops when removing a feature significantly decreases the OOB score, identified through a sliding window approach. This approach within 3DMASC combines filter-based pre-selection for interpretability with embedded selection to capture feature interactions within the RF model [273].

4.4.2 Semantic segmentation results

We performed both qualitative and quantitative evaluations of the semantic segmentation and change detection results. For the class-level evaluation, we used the Intersection over Union (IoU) as a metric. As a global indicator, we employed the overall accuracy (OA) and mean of the Intersection over Union (mIoU) [277]. The following equations were used to compute each of these metrics:

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP}} \tag{14}$$

$$\mathrm{OA} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{15}$$

$$\text{mean IoU} = \frac{\sum \mathrm{IoU_n}}{\mathrm{N}} \tag{16}$$

where TP, TN, FP, and FN refer to true positives and negatives, false positives, and negatives, respectively. N represents the total number of classes.

According to **Table 16**, which shows the results on HMLS subset, LightGBM demonstrates an overall accuracy (OA) of 0.94 positioned between the 3DMASC (RF) and KPConv approaches, which achieved OAs of 0.93 and 0.97, respectively. LightGBM also achieves a mean Intersection over Union (mIoU) of 0.60, demonstrating noteworthy proficiency in precisely delineating diverse semantic classes; it displays competitive efficacy across various categories, particularly in classes like Vegetation and Wires. However, KPConv outperforms both RF and LightGBM, particularly in classes like Ground, Rail, and the minority classes.

Table 16. Quantitative experimental results on the HMLS dataset.

Approach	OA	mIoU	Ground	Vegetation	Rail	Poles	Wires	Signaling	Fences	Installation	Building
KPConv	0.97	0.86	0.95	0.91	0.94	0.93	0.99	0.96	0.90	0.13	0.99
3DMASC	0.93	0.57	0.82	0.89	0.46	0.72	0.96	0.09	0.33	0.08	0.80
LightGBM	0.94	0.60	0.83	0.91	0.48	0.75	0.97	0.20	0.30	0.15	0.85

The results on SNCF subset show that the LightGBM method demonstrates an overall accuracy (OA) of 0.93 in the semantic segmentation task, positioned between the Random Forest (RF) and KPConv approaches, which achieved OAs of 0.91 and 0.97, respectively. LightGBM also achieves a mean Intersection over Union (mIoU) of 0.67, highlighting notable performance in accurately segmenting various semantic classes. It exhibits competitive performance across various semantic classes, with notable strengths in classes such as Rail. However, KPConv outperforms both RF and LightGBM method, particularly in classes like Ground and Signaling in **Table 17**.

Table 17. Quantitative experimental results on the SNCF dataset.

Approach	OA	mIoU	Ground	Vegetation	Rail	Poles	Wires	Signaling	Fences	Installation
KPConv	0.97	0.81	0.92	0.96	0.79	0.96	0.99	0.67	0.97	0.23
3DMASC	0.91	0.64	0.62	0.90	0.82	0.77	0.89	0.08	0.94	0.08
LightGBM	0.93	0.67	0.68	0.92	0.86	0.78	0.88	0.16	0.85	0.20

On the Infrabel subset, the LightGBM method demonstrates an overall accuracy (OA) of 0.97, confirming its position between the Random Forest (RF) and KPConv approaches, which achieved OAs of 0.96 and 0.99, respectively. LightGBM also achieves a mean Intersection over Union (mIoU) of 0.71, indicating significant performance in accurately segmenting various semantic classes. It exhibits competitive performance across a variety of semantic classes, with notable strengths in classes such as Vegetation and Wires. However, KPConv outperforms both the RF and LightGBM methods, particularly in classes like Poles and Installation in **Table 18**.

Table 18. Quantitative experimental results on the INFRABEL dataset.

Approach	OA	mIoU	Ground	Vegetation	Rail	Poles	Wires	Signaling	Fences	Installation
KPConv	0.99	0.84	0.99	0.84	0.95	0.97	0.99	0.40	0.69	0.89
3DMASC	0.96	0.70	0.96	0.84	0.85	0.88	0.99	0.22	0.56	0.28
LightGBM	0.97	0.71	0.97	0.86	0.87	0.89	0.98	0.25	0.63	0.26

Feature importance analysis revealed that the Normal dip (Dip Angle) at scales of 2 m emerged as the one of the most important features, indicating their significant role in the classification accuracy. Following these, verticality at 2 m scale was also identified as key contributor to the model's performance. These findings, visualized in our feature importance **Figure 55**, underscore the value of these specific features in enhancing the semantic segmentation of railway scene point clouds.

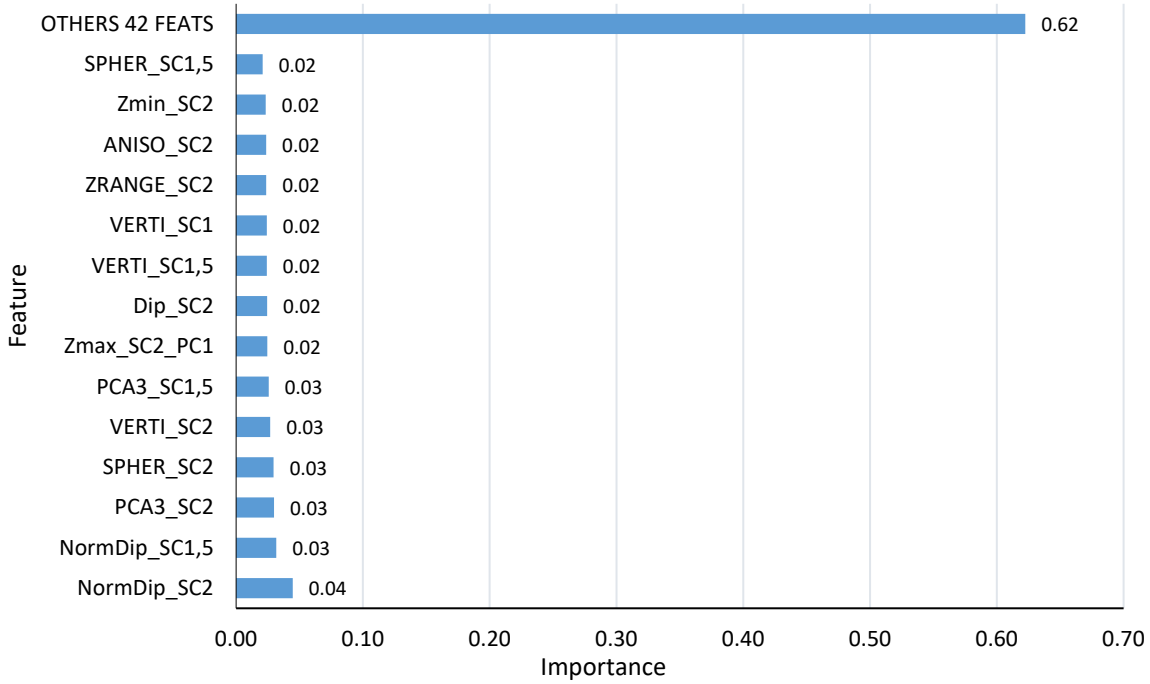


Figure 55. Feature importance selected using the LightGBM results on the Infrabel subset.

Since KPConv achieved the best performance results, we evaluated its generalization over other datasets. We conducted a series of additional experiments; we trained the model using only HMLS and assessed its performance on SNCF and INFRABEL datasets. Subsequently, we trained a larger model (KPConv**) utilizing all three datasets and evaluated its performance on the test subset. The results demonstrated a significant improvement in model generalizability when trained on a multi-context dataset. This highlights the importance of incorporating diverse data from various railway environments to develop models that can effectively handle real-world scenarios and adapt to different contexts. The results of these tests are summarized in the table below. The KPConv model trained only on HMLS gave results of 0.61 in mIoU on SNCF and 0.55 in mIoU on INFRABEL. This is mainly because the railway contexts, in general, are quite similar. But, the results were much improved when we used the whole dataset. We reached 0.76 mIoU for SNCF and 0.64 mIoU for INFRABEL. The bad result given by KPConv on the Fences class is due to its misclassification as a building even though it does not exist in INFRABEL. Also, KPConv** gave a result of almost 0 IoU in the Signaling class, as this feature was classified into the Pole class. **Table 19** summarizes these findings.

Table 19. Quantitative results of KPConv trained on HMLS and evaluated on SNCF and INFRABEL. KPConv ** was trained on all the datasets and evaluated on the test subset.

Dataset	Model	OA	mIoU	Ground	Vegetation	Rail	Poles	Wires	Signaling	Fences	Installation
SNCF	KPConv	0.95	0.61	0.88	0.94	0.61	0.84	0.89	0.26	0.81	0.24
	KPConv**	**0.97**	**0.76**	**0.91**	**0.96**	**0.78**	**0.97**	**0.99**	**0.83**	**0.98**	**0.47**
INFRABEL	KPConv	0.98	0.55	0.98	0.76	0.86	0.75	0.97	**0.27**	0	0.34
	KPConv**	**0.99**	**0.64**	**0.99**	**0.82**	**0.94**	**0.88**	**0.99**	0	**0.68**	**0.45**

KPConv training was monitored by tracking both training loss and accuracy (**Figure 56** and **Figure 57**). The training loss indicates the model's ability to learn from the provided data, decreasing as the model fits the training examples. Conversely, accuracy reflects the model's performance in correctly classifying points during training. A continuously decreasing loss accompanied by a rising accuracy towards a plateau suggests the model is effectively learning the semantic relationships within the point cloud data. Meanwhile, the loss ideally exhibits a continuous decrease as the model learns from the data; occasional spikes or fluctuations may occur.

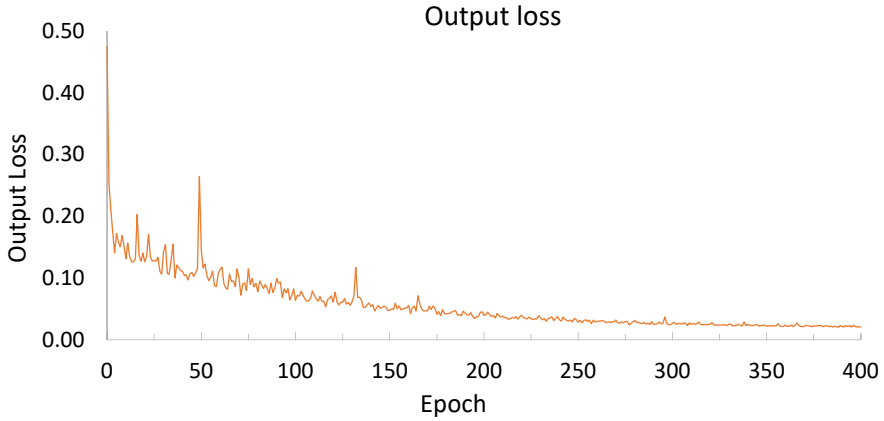


Figure 56. KPConv training loss on Rail3D Dataset (model named KPConv**).

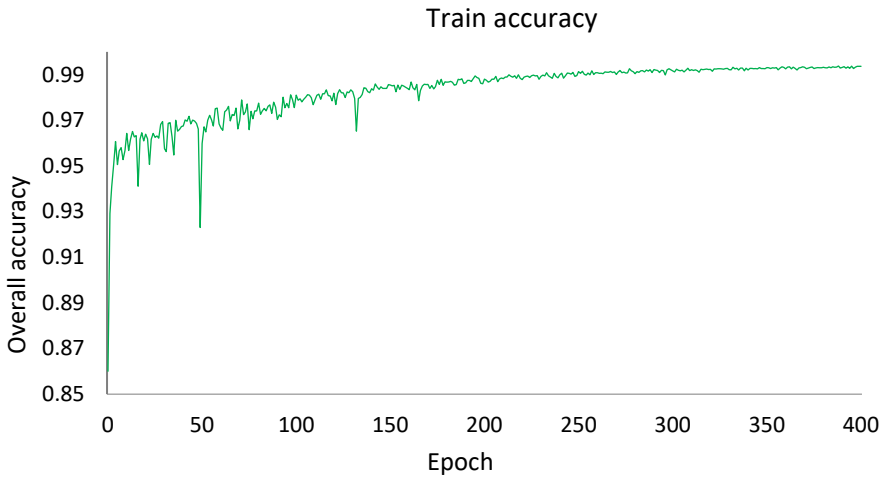
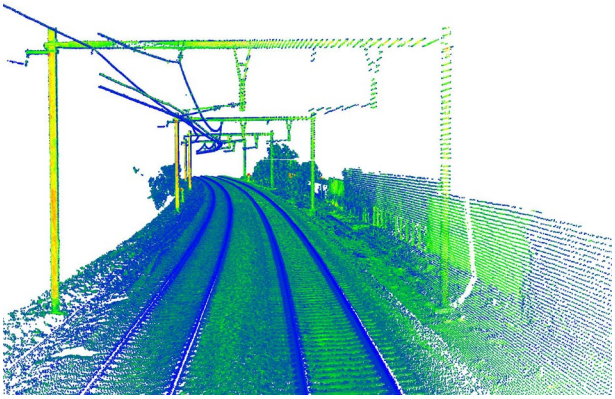


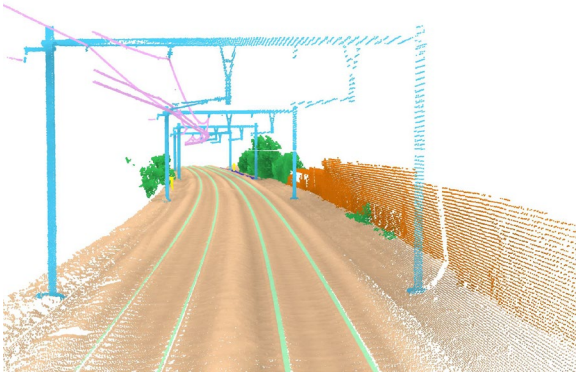
Figure 57. KPConv training accuracy on Rail3D dataset (model named KPConv**).

In the following figure we visually show samples of the point cloud, ground truth inputs, and the results of KPConv, LightGBM, and 3DMASC, successively.

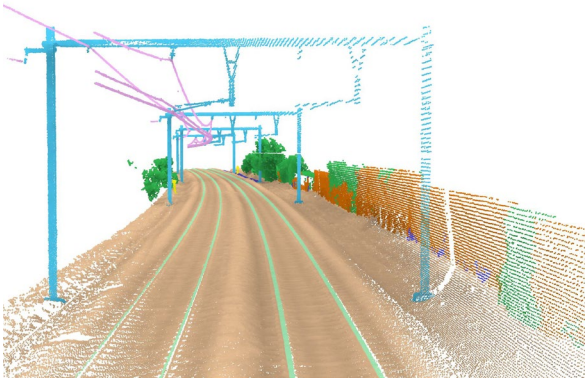
More qualitative results are presented in Appendix A for the HMLS subset, in Appendix B for the SNCF, and in Appendix C for Infrabel. Each include scenes, with the ground truth, KPConv predictions, KPConv errors, 3DMASC predictions, and 3DMASC errors. We have not visually identified any significant differences between 3DMASC and LightGBM, even though the latter gives better quantitative results. **Figure 63** highlights the problem of the misclassification of a fence next to railway tracks, which was classified as a building. This problem is due to the imbalance between these two classes, given that buildings are a minority but also that this type of fence can only be found in the INFRABEL dataset. All three approaches faced the same difficulty, with KPConv being the most accurate; visually, we cannot see difference between the 3DMASC and LightGBM results.



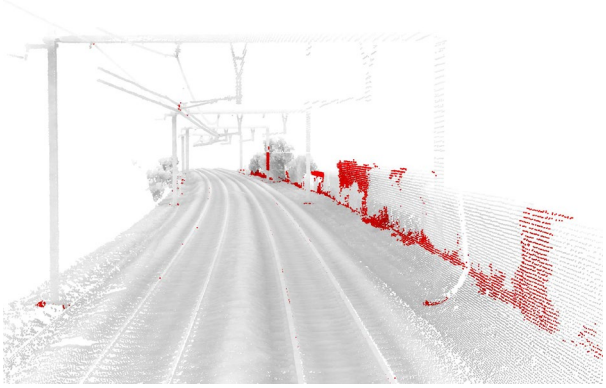
Input point cloud with intensity



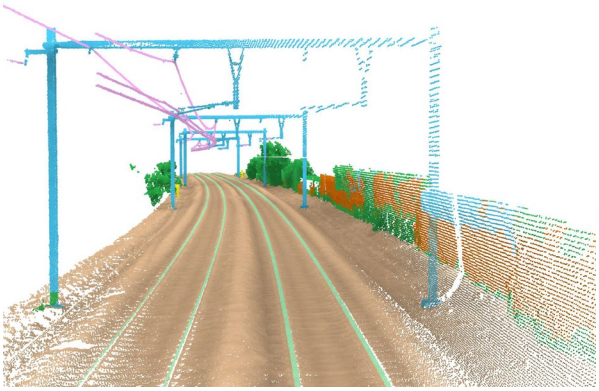
Ground truth



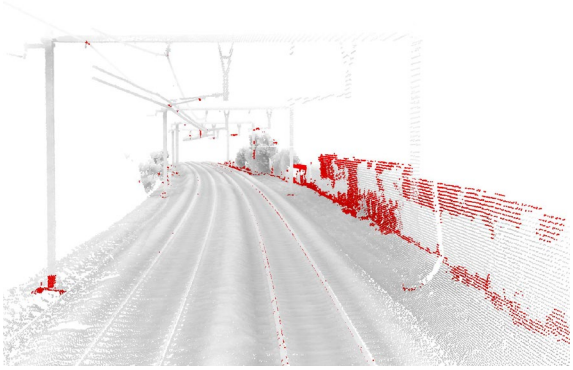
KPConv prediction



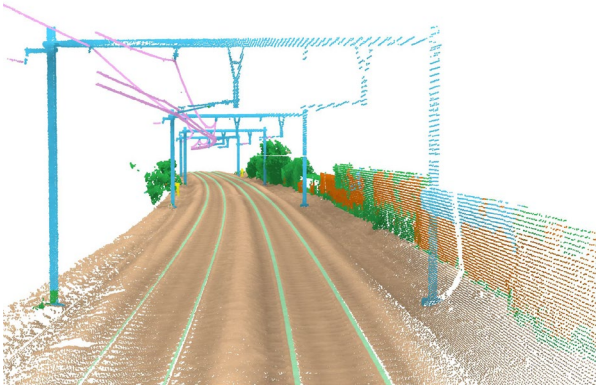
KPConv errors



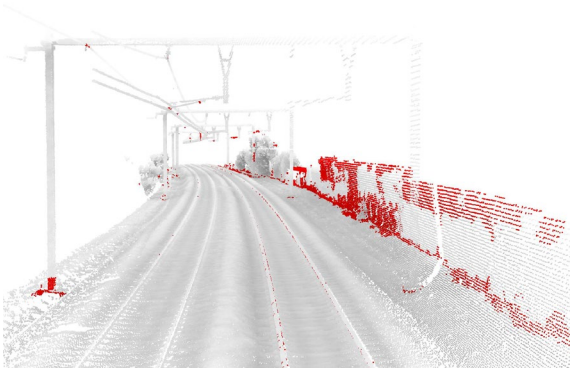
3DMASC prediction



3DMASC errors



LightGBM prediction



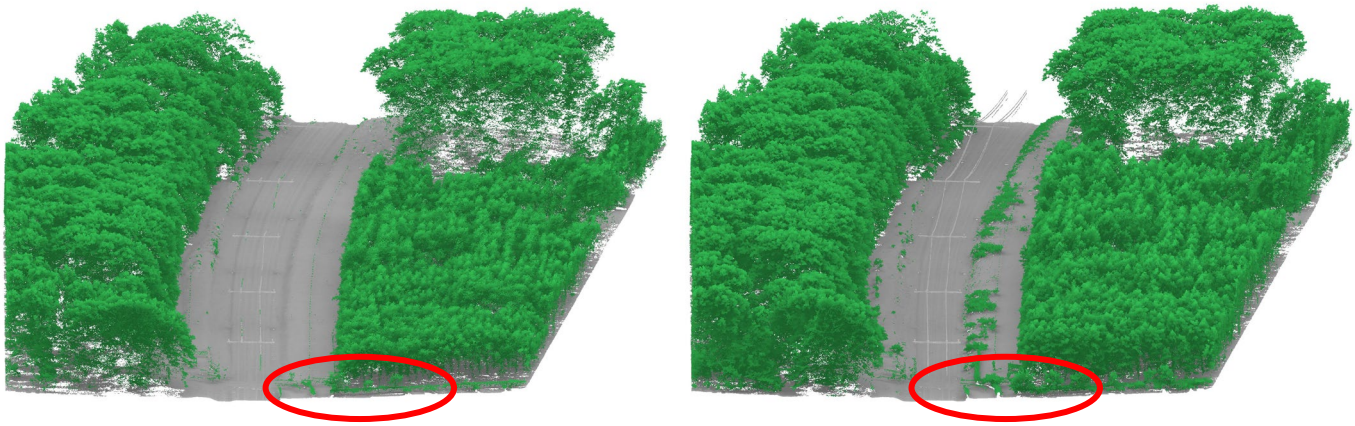
LightGBM errors

Ground	Vegetation	Rail	Poles	Wires	Signalling	Fences	Installation	Building

Figure 58. Input point cloud, with KPConv, 3DMASC, and LightGBM predictions. Each is associated with an error relative to the ground truth. The same class colors are used in the previous figures.

4.4.3 Change detection results

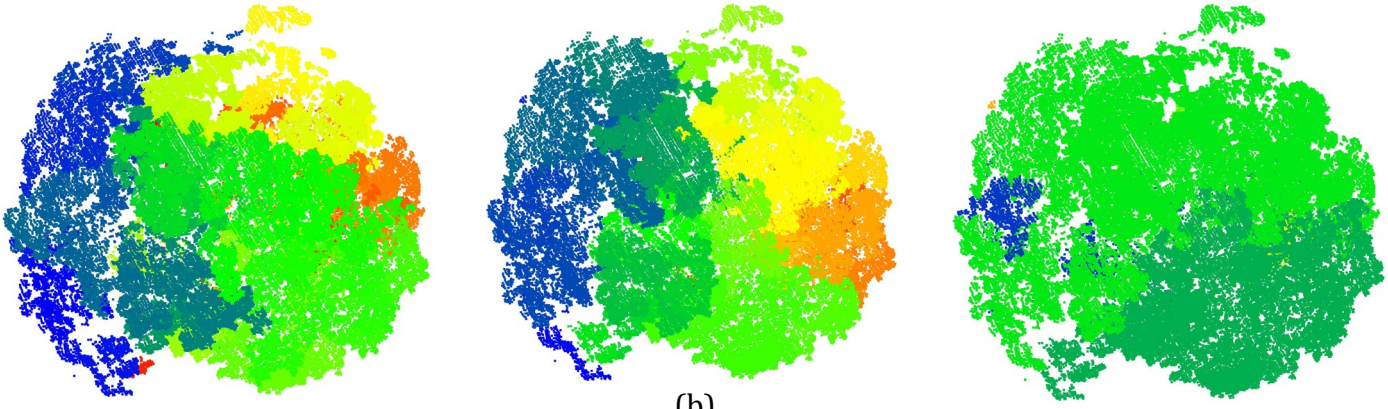
The semantic segmentation results on the data used for change task are globally consistent and visually coherent over the railway corridor. Vegetation is easy to identify due to its vertical structure, irregular surface, and spatial dispersion. As shown in **Figure 59**, most vegetated areas are correctly classified. However, small portions of railway infrastructure, such as poles or platform edges, were occasionally assigned to the vegetation class. These misclassifications are spatially limited and do not significantly affect the reliability of the subsequent object-based analysis.



(a) (b)

Figure 59. Semantic segmentation results on railway vegetation. In green we can see the vegetation class viewed using orthographic projection. The red ellipse shows misclassification.

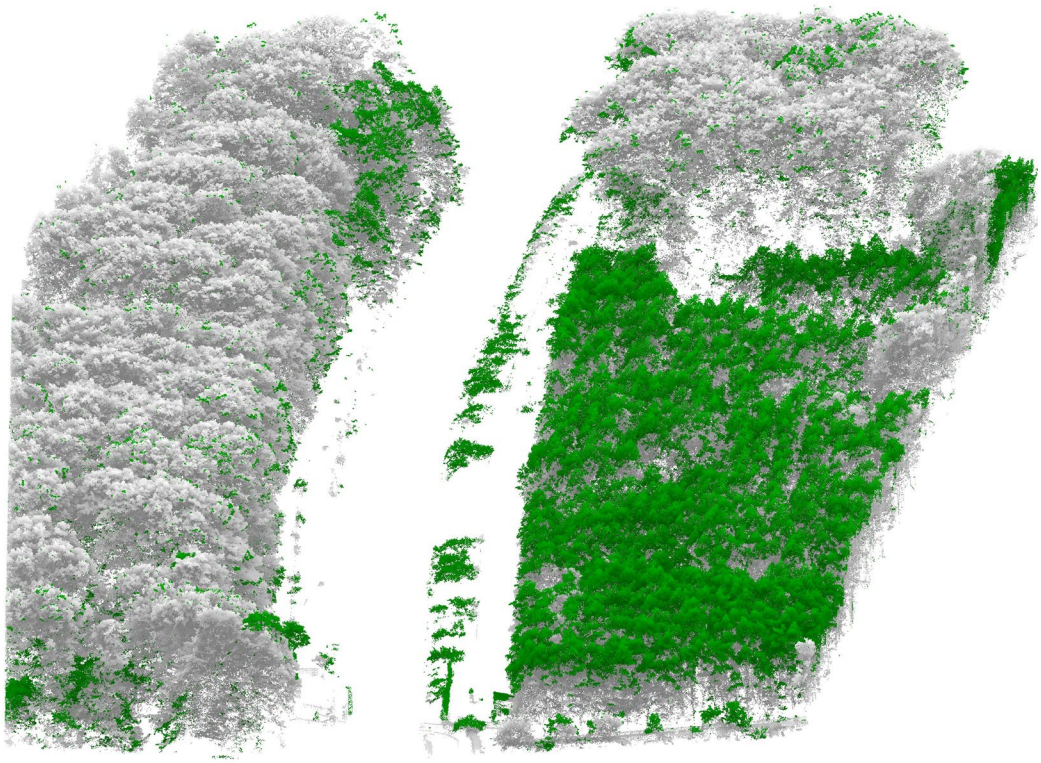
The figure below (**Figure 60**) illustrates the outcome of hierarchical vegetation segmentation at three scales on the 2023 dataset. The fine level (a), with 545 clusters, reflects over-segmentation where individual vegetation structures (e.g., trees or dense shrubs) are separated into small, detailed units. The mid-level (b), with 114 clusters, achieves a balance, aggregating points into semantically meaningful tree crowns or vegetation patches. The coarse level (c), with only 14 clusters, groups larger vegetation masses, suitable for broader monitoring purposes but potentially merging distinct botanical entities.



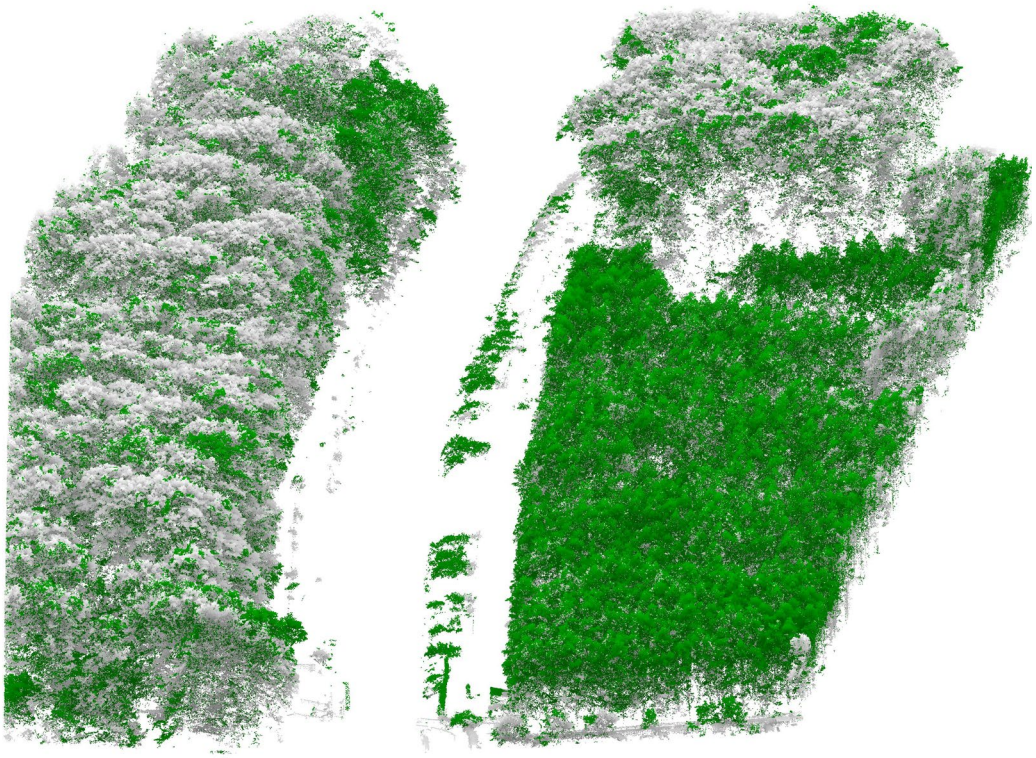
(a) (b) (c)

Figure 60. Multi-scale vegetation clustering. (a) fine-level 545 clusters, (b) mid-level 114 clusters, (c) coarse-level 14 clusters.

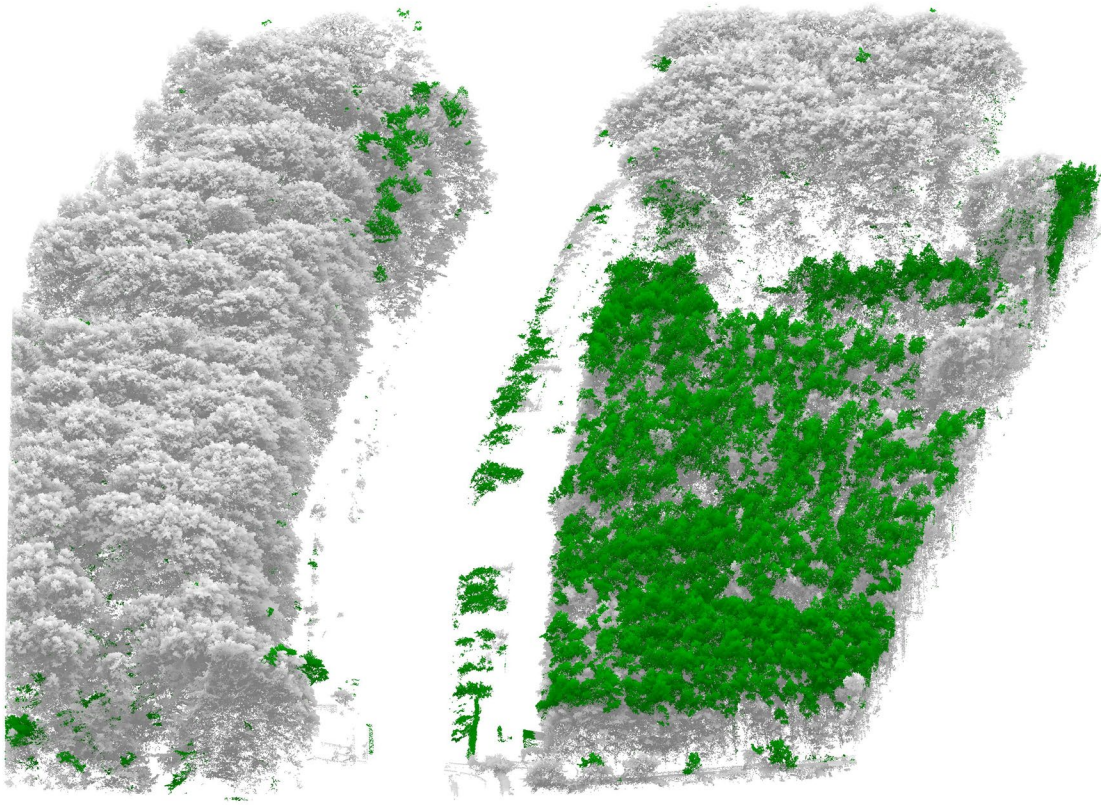
To evaluate the effectiveness of the proposed multi-scale object-based change detection method, we present in **Figure 61** a comparison with two widely used point-based methods: Cloud-to-Cloud (C2C) and Multiscale Model-to-Model Cloud Comparison (M3C2). To provide a more interpretable comparison, we isolated a test zone and applied our method across three segmentation levels (**Figure 62**). The fine scale successfully identifies subtle changes, especially in small vegetation clusters. The mid-level, due to larger object granularity, reveals more extensive change regions, where any unconfirmed similarity leads to the entire object being flagged as changed. Finally, the coarse scale exaggerates change extent, with entire vegetation blocks marked as changed. This over-detection is attributed not only to actual vegetation evolution but also to segmentation inconsistencies arising from variations in acquisition parameters and point density across epochs.



(a). cloud to cloud comparison results (1 meter threshold)



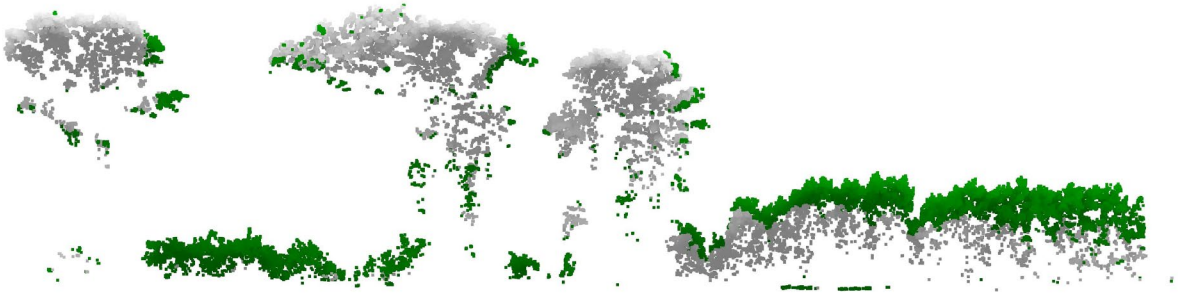
(b). Multiscale model to model cloud comparaison



(c). Results of our fine-scale object-based method

Figure 61. Visualization of change detection results. Green colors show the changed area with a 1m threshold for (a). cloud to cloud, (b) multi scale model to model comparison, (c) our approach.

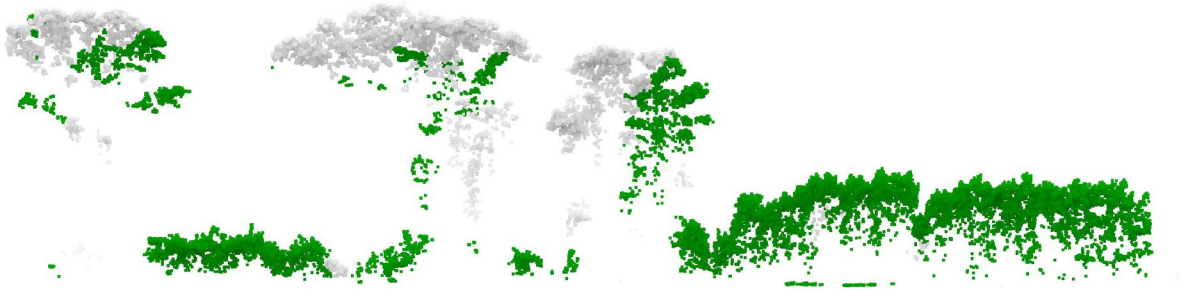
(a). Reference point clouds (2018: gray, 2023: green)



(b). C2C change detection (green)

(c). M3C2 change detection (green)

(d). Fine-scale vegetation change (green)



(e). Mid-scale vegetation change (green)

(f). Large-scale vegetation change (green)

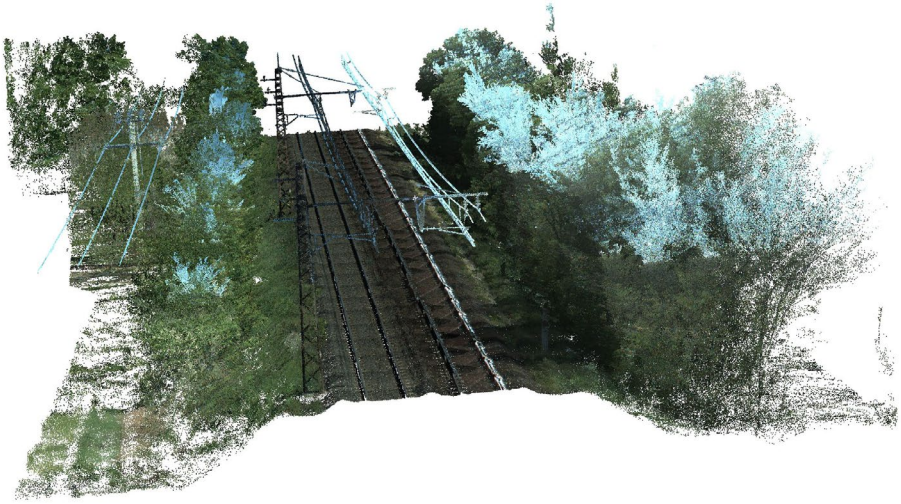
Figure 62. Vertical sections for the comparison of point-based (C2C, M3C2) and object-based change detection at three segmentation scales (fine, medium, coarse)

4.5 Discussion

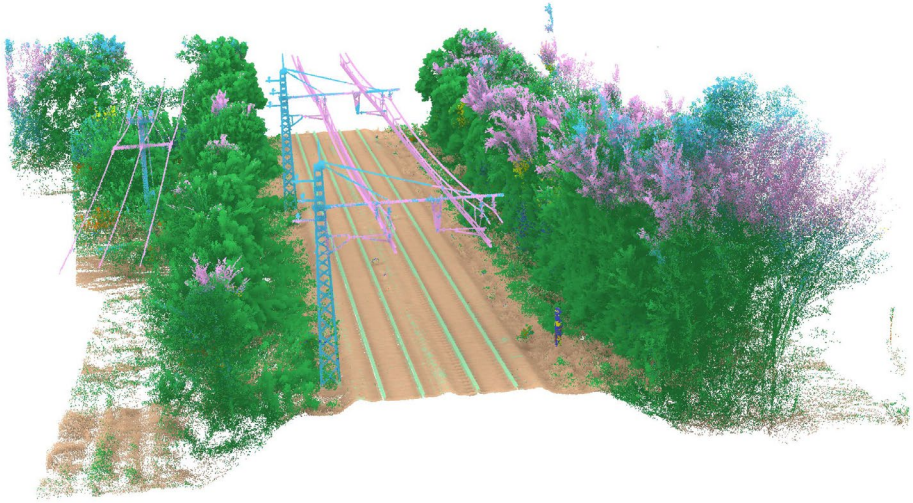
This work addresses three complementary aspects: firstly, the proposal of four subset representing different contexts of railway environments; secondly, the introduction of three baselines to address semantic segmentation challenges; and thirdly, the design of a robust method for vegetation change detection based on multi-level segment analysis and object-level matching. Regarding the proposed dataset, it presents the first comprehensive dataset for semantic seg-mentation in railway environments, covering three distinct railway contexts from Hungary, France, and Belgium. These datasets provide rich resources for applications in railway environments. With a significant annotation of over 288 million points, the created datasets stand out in both size and diversity compared to existing datasets. This extensive volume of annotated points enhances the dataset's ability to cover a wide range of situations and variations within railway environments. Furthermore, the datasets developed across these railway environments ensure effective learning for machine learning models, confirming their generalization to different contexts. This was confirmed by the results obtained in **Table 19**.

The comparison between KPConv-, 3DMASC-, and LightGBM-based results provides valuable insights about the performance of different classification methods for large-scale rail borne point cloud data. The results of the comparison indicate that, while KPConv may excel in capturing intricate local geometric features, LightGBM highlights promising performance in handling large-scale datasets with its computational efficiency. 3DMASC, although a reliable method, may exhibit limitations in dealing with the complexity and unbalanced nature of railways point cloud data. Furthermore, LightGBM offers superior computational efficiency and faster training times compared to traditional Random Forests. The performance of different methods was assessed through a combination of qualitative and quantitative evaluations. Across **Table 16**, **Table 17**, **Table 18**, the LightGBM method consistently gave good performance in semantic segmentation, with overall accuracies (OAs) ranging from 0.93 to 0.97 and mean Intersection over Union (mIoU) values from 0.60 to 0.71. Notably, the LightGBM approach highlights efficient results within a brief timeframe, 100 times faster than 3DMASC, emphasizing its practical applicability. This is because LightGBM is designed to be highly parallelizable while 3DMASC is not. Parallelization allows LightGBM to divide the training task into multiple smaller tasks, which can be run on multiple cores or CPUs simultaneously. While demonstrating competitive accuracy, particularly excelling in classes like Rail, Vegetation, and Wires, the machine learning method consistently falls slightly behind the performance of KPConv. The latter's superior accuracy, especially in specific classes, suggests areas for potential refinement in the handcrafted feature method, highlighting the ongoing challenge of balancing speed and precision in semantic segmentation tasks. The qualitative results obtained using various methods have confirmed the quantitative findings. The traditional ML methods yielded qualitative results closely aligned with the ground truth across all three subsets. This underscores the good selection of geometric features, indicating a positive impact on differentiating railway objects. While color and intensity are often valuable features that can enhance classification performance, we opted to exclude them from our study for

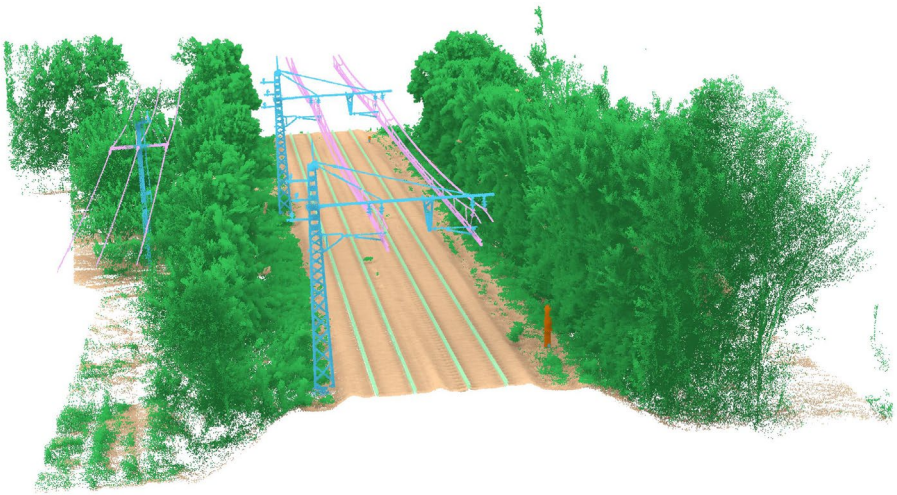
two primary reasons. Firstly, the color data acquired exhibited inferior quality due to variations in acquisition times and seasons, leading to heterogeneity between contexts. Secondly, the intensity values lacked calibration across the three contexts, as different lidar sensors were employed. To validate our choice of relying exclusively on spatial coordinates (x, y, and z) as input, we conducted a test using KPConv that verified the adverse effect of color and intensity on classification accuracy within our specific case study (refer to **Figure 63**). One solution to correct this sky-colored points effect would be to perform a pre-processing step, as suggested in [278].



(a)



(b)



(c)

Figure 63. The poor quality of the colors led to poor classification (e.g., in vegetation). (a) input point cloud, (b) predictions with colors, (c) predictions without colors.

In addition, misclassification occurred where signaling poles were mistakenly categorized as catenary poles. This is due to the strong resemblance between the two and the lack of use of intensity or a specific color or geometric shape for these signs. In other cases, as shown in **Figure 58**, LightGBM and 3DMASC were unable to extract the Fence class correctly, given the strong similarity with the Building class. However, KPConv was able to classify it correctly. This led to a further experiment on KPConv to assess its generalizability. The use of a multi-context dataset, encompassing railways from three different countries, was shown to be crucial for boosting the generalizability of semantic segmentation models. By training on data from diverse railway environments, the model develops a more robust representation of railway structures and can better adapt to unseen contexts. This was demonstrated in our experiments, where the KPConv model trained on our multi-context dataset achieved significantly higher mIoU scores (0.76 and 0.64) on the SNCF and INFRABEL datasets compared to the model trained on the HMLS dataset alone (0.61 and 0.55). This suggests that our dataset can generalize to unseen railway environments, making it a valuable resource for developing robust semantic segmentation models for railway applications.

The object-based 3D change detection approach was applied after the semantic segmentation phase, specifically targeting vegetation. The method relies on a multiscale segmentation strategy applied separately to the two epochs. Three levels of segmentation granularity were considered: fine, intermediate, and coarse. For each level, vegetation objects were described by geometric features (centroid, 95th height percentile, and eigenvalues of the covariance matrix) and matched using a spatial radius constraint and feature similarity.

Figure 62 illustrates a comparative analysis between our method and two established point-based algorithms: C2C and M3C2. Visually, C2C produces dense and noisy change maps due to its pointwise nature and sensitivity to sampling variation. M3C2 partially

mitigates this issue by considering surface normals and local roughness but remains sensitive to vegetation density and occlusion. In contrast, our multiscale object-based method provides spatially aggregated and geometrically interpretable results. At fine scale, small vegetation structures are captured, enabling the detection of localized growth or removal along railway edges. At coarser scales, broader patterns such as canopy expansion or bulk vegetation clearance are emphasized, offering a scalable view depending on the monitoring need.

The method has several advantages. First, it reduces the impact of acquisition discrepancies by operating at the object level, rather than individual points. Second, the use of feature-based matching avoids reliance on exact segment ID propagation, which is unreliable when the segmentation is performed independently per epoch. Third, the multiscale strategy allows adaptation to different change types and operational resolutions. However, some limitations remain. The method depends on the quality of the segmentation and the geometric distinctiveness of the objects, which can vary with vegetation density.

4.6 Conclusions

This chapter introduced two core contributions addressing semantic segmentation and object-based change detection in the context of railway environments.

The first contribution concerns the creation of the Rail3D dataset, designed to support supervised learning for semantic segmentation across multi-context railway scenes. With annotated point clouds from Hungary, France, Netherlands, and Belgium, the dataset captures a variety of acquisition conditions and object classes relevant to infrastructure monitoring. Comparative experiments demonstrated that KPConv, when fine-tuned, outperforms other tested models in terms of segmentation accuracy, achieving a mean Intersection over Union of 86%. LightGBM provides a viable alternative when computational constraints are present, though with reduced accuracy. These results validate the robustness of deep learning approaches in capturing class boundaries under variable acquisition geometries and highlight the relevance of Rail3D as a benchmark for future research.

The second contribution introduced a novel multiscale object-based methodology for detecting vegetation changes from bi-temporal airborne LiDAR data. Vegetation points were extracted independently for each epoch using KPConv, then structured hierarchically into fine, intermediate, and coarse-scale segments. For each level, objects from the recent epoch were matched to previous ones based on their spatial features: centroid position, height percentile (Z95), and shape descriptors (eigenvalues of the covariance matrix). Matching was refined using a combination of spatial radius filtering and similarity comparison. Changes were declared only when no similar object was found in the previous epoch. This hierarchical approach was evaluated through visual comparison against classical point-based methods (C2C and M3C2), highlighting that our approach reduces false positives and better captures meaningful vegetation growth or disappearance at object level.

The advantages of the proposed method include its robustness to noise, improved interpretability through object-level structuring, and operational alignment with how railway managers monitor encroachment. However, its effectiveness depends on the quality and granularity of segmentation, which can be influenced by point cloud density differences between epochs. Visual inspection was preferred over metric comparison to assess change relevance across multiple scales, particularly for vegetation structures that are challenging to evaluate using point-based distances alone. Together, these contributions partially address the third research question of this thesis..

CONCLUSION AND PERSPECTIVES

Findings and contributions

This section synthesizes the key contributions made in this thesis and provides answers to the research questions introduced at the beginning. It reflects on how the proposed methodologies, implementations, and experimental results collectively advance the state of the art in 3D point cloud change detection, object-based methods, semantic enrichment, and structured urban modeling. The findings are articulated through the lens of the research questions, each answered with a discussion of the developed approaches and their relevance in practical and theoretical contexts.

The central aim of this thesis was to explore how 3D point cloud data, collected across different time epochs, can be used to automatically detect, classify, and structure changes in urban and infrastructure environments. To achieve this goal, this work contributes to bridging the gap between geometric analysis, and semantic segmentation in a modular and extensible pipeline. It provides insights not only into methodological aspects but also on the organizational structuring of change detection outputs into interoperable formats, essential for real-world applications such as digital twins, monitoring systems, and urban reconstruction.

The first research sub-question addressed was: **How can we reliably detect and characterize changes in 3D environments using multi-temporal point cloud data?** (Answers in Chapter 01).

To answer this, a detailed and critical review of existing 3D change detection methods was conducted, providing insights into their capabilities, limitations, and practical challenges. Chapter 1 explored various methodological categories, including pixel-based, point-based, voxel-based, and object-based methods, evaluating their performance in handling issues like data heterogeneity, occlusions, and spatial variability commonly encountered in real-world datasets. This review identified that object-based methods and integrating semantic segmentation hold significant potential due to their interpretability and structured output capabilities. It highlighted that reliable detection and characterization of changes could benefit significantly from incorporating semantic context alongside geometric descriptors, setting the foundation for the subsequent developments in the thesis.

The second research sub-question asked: **How can semantic and geometric features be fused at the object level to improve change classification accuracy?** (Answers in Chapter 02)

This question was directly addressed in Chapter 2, where an integrated pipeline combining semantic segmentation and geometric analysis at the object level was developed. The proposed method involved initial semantic segmentation of the point cloud data, assigning labels such as buildings, vegetation, moving objects, and ground, followed by clustering into distinct clusters. These segments were then matched across epochs using geometric proximity and semantic class consistency. Experimental

evaluation using a simulated urban dataset showed that integrating semantic and geometric features significantly enhanced the robustness of object matching and change classification. The object-level approach effectively managed complex scenarios like partial occlusions, viewpoint differences, and noise, which are challenging for purely geometric or semantic approaches alone.

The third research sub-question investigated: **How can detected changes be formalized into structured, interpretable, and interoperable models suitable for urban applications?** (Answers in Chapter 03 and Chapter 04)

To operationalize detected changes, a structured data model was used and implemented that transforms raw change observations into a machine-readable urban model in chapter 3. A key innovation of this thesis is the structuring of change into the CityJSON format, leveraging its lightweight and object-based characteristics. Detected changes are no longer treated as simple point-level anomalies but instead translated into structured features linked to specific semantic objects. A dedicated extension to CityJSON was designed and implemented to encapsulate change metadata, object identity, timestamps, and change types. This not only makes the change data interpretable and traceable but also ensures compatibility with downstream tools for visualization, analysis, and integration into broader urban information systems. The structuring process marks a decisive step toward interoperable and sustainable use of point cloud-based change detection in real-world applications such as building monitoring, vegetation dynamics, and infrastructure management, as seen in chapter 4. This rule-based framework enables the system to work even in the absence of labeled data, relying instead on object relations, semantic constraints, and change rules defined by the user. The result is a replicable and extensible approach to modeling dynamic urban environments with minimal supervision.

Together, these contributions represent a structured response to the main research question posed at the beginning of this thesis. They build a coherent path from raw bi-temporal point cloud data to structured semantic change models, anchored in a rigorous fusion of geometric, and semantic features. While each research question was addressed through a dedicated methodological block, the strength of the proposed approach lies in its integrated and object-based design, aligning with current needs for modular, interpretable, and extensible solutions in urban 3D modeling and change monitoring. The methodology developed throughout this thesis demonstrates that it is possible to move beyond isolated pointwise detection and toward a higher level of abstraction where changes are identified, classified, and structured in a way that aligns with real-world entities.

Research extensions

This section presents the potential research directions that emerge from the findings and contributions of this thesis. These directions aim at improving our work on change detection from bi- and multi-temporal point clouds. The perspectives detailed below are closely aligned with the challenges and limitations discussed in the previous chapters and aim to bridge the remaining gaps in current practices. Below, we present six relevant research directions that will have a major impact on this research topic.

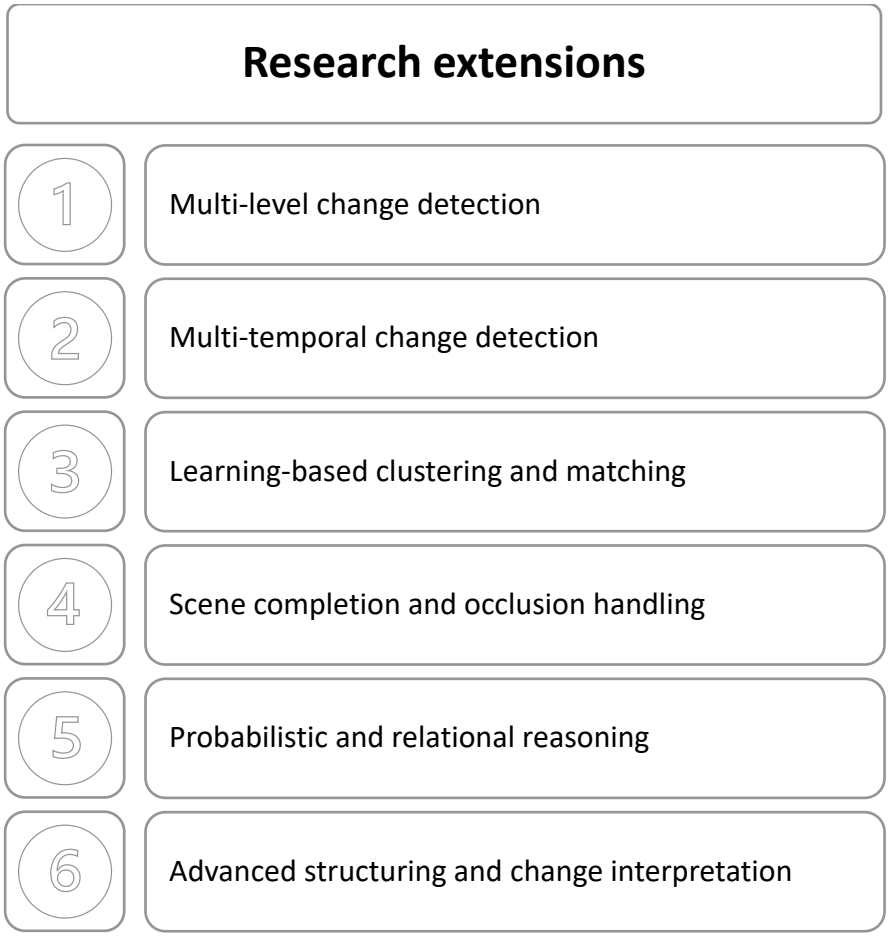


Figure 64. Summary of the research extensions based on our research results and contributions.

1. Toward multi-level change detection

The methods developed in this thesis focus on object-level change detection by leveraging semantic segmentation and geometric comparison between two epochs. While this approach proved effective for generating interpretable and structured output, especially for building and vegetation monitoring, it also revealed several limitations due to its reliance on a single level of abstraction. As demonstrated in **Chapter 2**, errors in object-level change classification often originate from earlier stages in the processing chain, such as segmentation inaccuracies or local point cloud artifacts. In Chapter 4, changes in vegetation were frequently under- or over-estimated due to coarse or fine segments. These observations highlight the need for a more flexible and hierarchical framework that enables change detection across multiple spatial levels (points, segments, and objects) and accounts for the interdependencies between them. For instance, detecting raw deviations at the point level, aggregating them at the segment level, and interpreting them semantically at the object level would support both fine-grained and high-level understanding.

Moreover, incorporating temporal granularity would further strengthen change interpretation. Rather than relying solely on discrete comparisons between two time points, multi-scale temporal modeling (tracking progressive, abrupt, or cyclic

transformations) would allow the system to adapt to different application needs, such as short-term incident detection or long-term urban evolution analysis. This is particularly relevant in the context of multi-epoch datasets, as discussed in the new research direction on multi-temporal monitoring.

2. Towards multi-temporal change detection

The methods developed in this thesis focus on bi-temporal change detection, which enables the identification of discrete transformations between two epochs. However, many real-world processes (such as vegetation growth, maintenance cycles, or gradual structural alterations) are inherently continuous or periodic. This was particularly evident in **Chapter 4**, where vegetation changes along railway corridors were modeled as binary events, limiting the system's ability to capture intermediate growth stages or recurrent interventions. Extending the framework to incorporate multi-temporal point clouds offers a more comprehensive understanding of scene dynamics by linking detected changes to temporal patterns. Such an approach would allow tracking of progressive overgrowth, recovery after clearing, or seasonal variation, while improving robustness to acquisition inconsistencies and noise. Beyond vegetation, it could support dynamic monitoring of infrastructure (powerline monitoring), construction sites, or natural phenomena. Embedding temporal reasoning into the change detection pipeline would contribute to research questions 1 and 3, by enabling more reliable characterization of change and its integration into structured, time-aware models. Future work could explore spatio-temporal graph representations, temporal feature learning, or recurrent neural architectures to formalize the evolution of objects over time and inform predictive applications.

3. Learning-based clustering and object matching

This thesis introduced an object-based change detection framework relying on the segmentation of 3D point clouds using Cut-Pursuit and the matching of segmented objects via nearest-neighbor search in a geometric descriptor space. While this approach achieved promising results in relatively structured environments, its performance proved sensitive to geometric variation, partial occlusions, and segmentation inconsistencies. As observed in **Chapter 2**, the classification of objects as “disappeared” or “modified” was occasionally misled by incomplete. Such errors highlight the limitations of relying solely on handcrafted geometric features for matching, especially when dealing with mobile mapping data where point density, occlusions, and acquisition angles vary between epochs.

To address these limitations, future work should explore learning-based clustering and object matching strategies. By replacing hand-engineered descriptors with deep features learned from data, it becomes possible to capture more robust and semantically meaningful object representations. Networks designed for point cloud encoding could be adapted to extract embeddings for object-level segments, facilitating more accurate correspondence across time. Furthermore, learning-based models can be trained to tolerate geometric noise or partial object visibility, which are frequent in real-world railway or urban environments.

4. Scene completion and occlusion handling

One of the central challenges encountered throughout this thesis lies in the limited completeness of mobile laser scanning data, which often suffers from occlusions and geometric holes due to the constraints of sensor positioning and the urban environment itself. This issue is particularly problematic for change detection tasks, as missing data may result in the false attribution of changes or the failure to detect subtle modifications. Although the object-based pipeline developed in this thesis offers robustness in handling partially observed objects, its performance remains constrained in complex scenes, such as densely vegetated areas or under urban structures like bridges.

A promising research direction involves integrating scene completion techniques into the change detection workflow to recover plausible geometry in occluded or sparsely sampled regions. Recent advances in semantic scene completion (SSC) networks, such as SCPNet [279], have shown that multi-path architectures and multi-frame knowledge distillation can improve voxel-wise predictions in both geometry and semantics. Building on these ideas, future work could explore the incorporation of class-aware completion strategies, where completion is adapted based on the semantic category of the object (e.g., planar interpolation for rooftops, volumetric estimation for vegetation, or symmetry priors for poles and facades). This would require learning class-specific priors that can guide the inference of missing geometry in a semantically meaningful way. Furthermore, the notion of temporal consistency could be introduced by leveraging multi-epoch data to inform scene completion, thereby distinguishing between true change and occlusion artifacts [280]. In particular, scenes that are incomplete in one epoch but well captured in another offer a unique opportunity to model and propagate geometry across time.

Finally, embedding completion mechanisms directly within the segmentation and change detection architectures, such as through joint training objectives or shared feature representations, could enable end-to-end optimization [281]. This approach would reduce the reliance on separate preprocessing stages and enable better exploitation of contextual cues, especially in environments characterized by repetitive structures or partial visibility. Integrating these advancements into the current pipeline would address a major limitation identified in **Chapter 3** and **Chapter 4** and open the door to more robust and interpretable 3D change detection in real-world applications.

5. Probabilistic and relational reasoning over changes

Although this thesis primarily adopts a deterministic approach to change classification (discrete labels such as “appeared,” “disappeared,” or “modified”), the complexity and ambiguity inherent in real-world 3D data call for a more flexible and interpretable framework. As observed in **Chapter 3**, change decisions are often based on geometric thresholds and object comparisons, which, while effective in clear cases, may lead to overconfident conclusions when the input data is incomplete, noisy, or poorly segmented. In such cases, false positives and misclassifications are particularly common around scene borders, dense vegetation, or partially scanned structures. This motivates a shift toward probabilistic reasoning, where change predictions are not treated as

binary outcomes but are instead associated with confidence scores that reflect the reliability of the evidence. Such probabilistic outputs could inform downstream processes, such as manual validation, infrastructure monitoring, or risk-aware decision-making.

In parallel, the current framework models changes independently at the object level, without considering inter-object relationships or spatial context. However, as seen in the results from both urban and railway scenes, many real-world transformations involve coordinated or causally related events, for instance, vegetation clearance preceding infrastructure installation, or building demolition followed by road reconfiguration. Capturing these spatial and semantic relations between changes can greatly enhance the interpretability of the outputs and support higher-level reasoning about urban dynamics. A promising direction would involve representing the scene and its evolution as a spatio-temporal graph, where nodes correspond to objects and edges encode spatial proximity, semantic similarity, or functional relationships. Such graph-based structures could serve as a substrate for both uncertainty modeling and relational inference. For instance, belief propagation techniques or probabilistic graphical models could be used to refine change hypotheses based on the behavior of neighboring objects. Incorporating relational constraints (such as object co-occurrence, semantic compatibility, or temporal consistency) would further reduce isolated misclassifications and improve the coherence of the change detection results. These approaches would directly contribute to research questions 2 and 3 by enabling the joint interpretation of geometric, semantic, and contextual information in a unified reasoning framework.

In the long term, formalizing change detection within a knowledge representation system, possibly guided by domain-specific ontologies (e.g., for railway assets or urban planning), would enable explainable and queryable interpretations of scene evolution. This would allow stakeholders to not only detect that a change occurred, but to understand why, how, and in relation to what. Such capabilities would transform the current pipeline from a descriptive tool into a more intelligent and context-aware system for 3D spatial analysis and digital twin integration.

6. Advanced structuring and change integration in city models

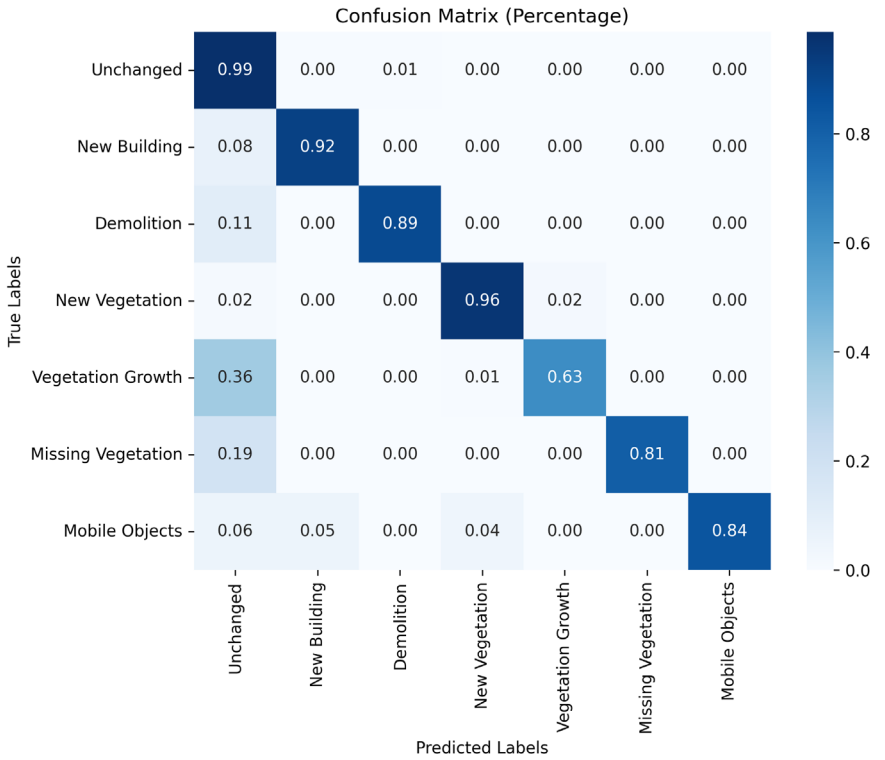
The thesis's **Chapter 3** focused on the generation of CityJSON models enriched with detected changes. This structured output format offers a valuable bridge between point cloud analysis and urban digital twin applications by enabling semantic integration, object formalization, and compatibility with existing 3D GIS platforms [282]. However, as demonstrated in Chapter 3, the current structuring approach is rule-based and limited to a subset of semantic classes (e.g., buildings and vegetation), relying on hardcoded logic to encode changes such as additions or removals. While effective for controlled use cases, this approach lacks scalability, flexibility, and generalization to more diverse urban objects or complex temporal scenarios.

Future research should aim to extend and automate the semantic structuring process, moving beyond static rules toward methods that can infer structure from geometric and contextual patterns. One important direction lies in the formal linkage of multi-epoch changes to persistent object identifiers, enabling object-based tracking across time and

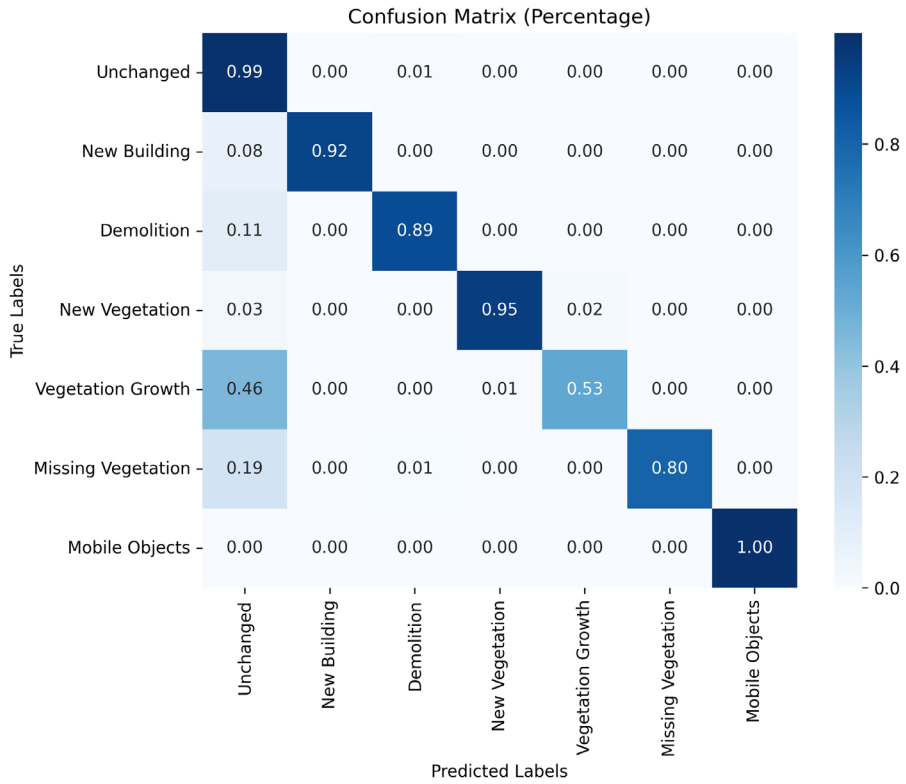
the generation of structured change logs that record what changed, when, and under what conditions. This capability would be essential for applications such as infrastructure lifecycle monitoring, permit verification, or urban growth modeling. Moreover, aligning the enriched models with external data sources (e.g., cadastral records, simulation outputs) would reinforce their operational relevance.

In addition to the core extensions proposed, a valuable direction is to explore the applicability of the developed methods beyond the specific contexts addressed in this work. Future research should investigate how the proposed semantic-geometric change detection framework can be adapted to other domains such as forestry, cultural heritage conservation, environmental monitoring, and industrial asset inspection. Furthermore, extending the methodology to accommodate different object types (e.g., linear infrastructures, utility networks, or natural elements like rock formations) can reveal the flexibility of the approach across diverse application scenarios.

APPENDIX A



Confusion matrix of the object-based change detection



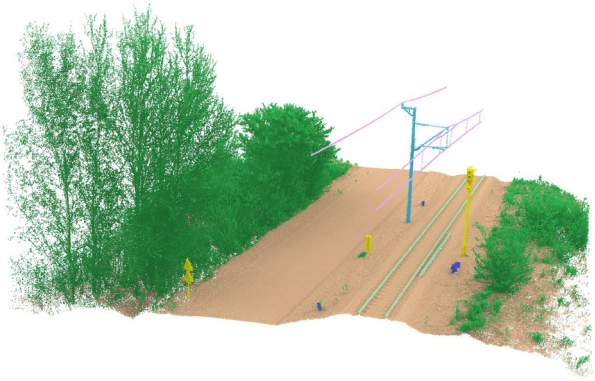
Confusion matrix of the point-based change detection

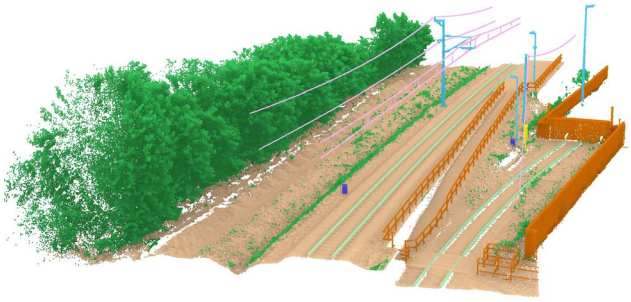
Figure A. Comparison between the confusion matrix: (a) object-based; (b) point-based.

APPENDIX B

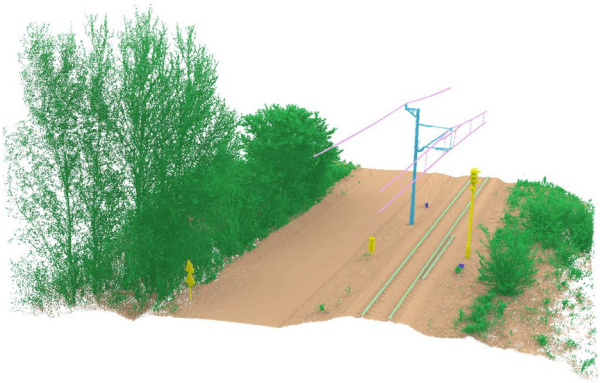
Input

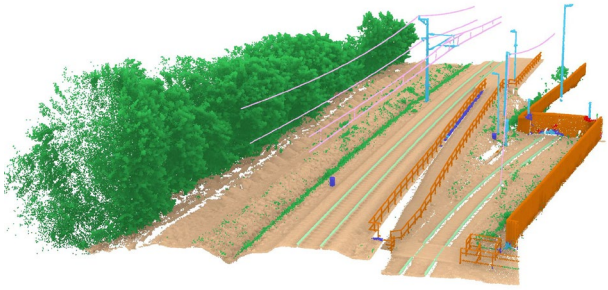
Ground Truth





KPConv Prediction





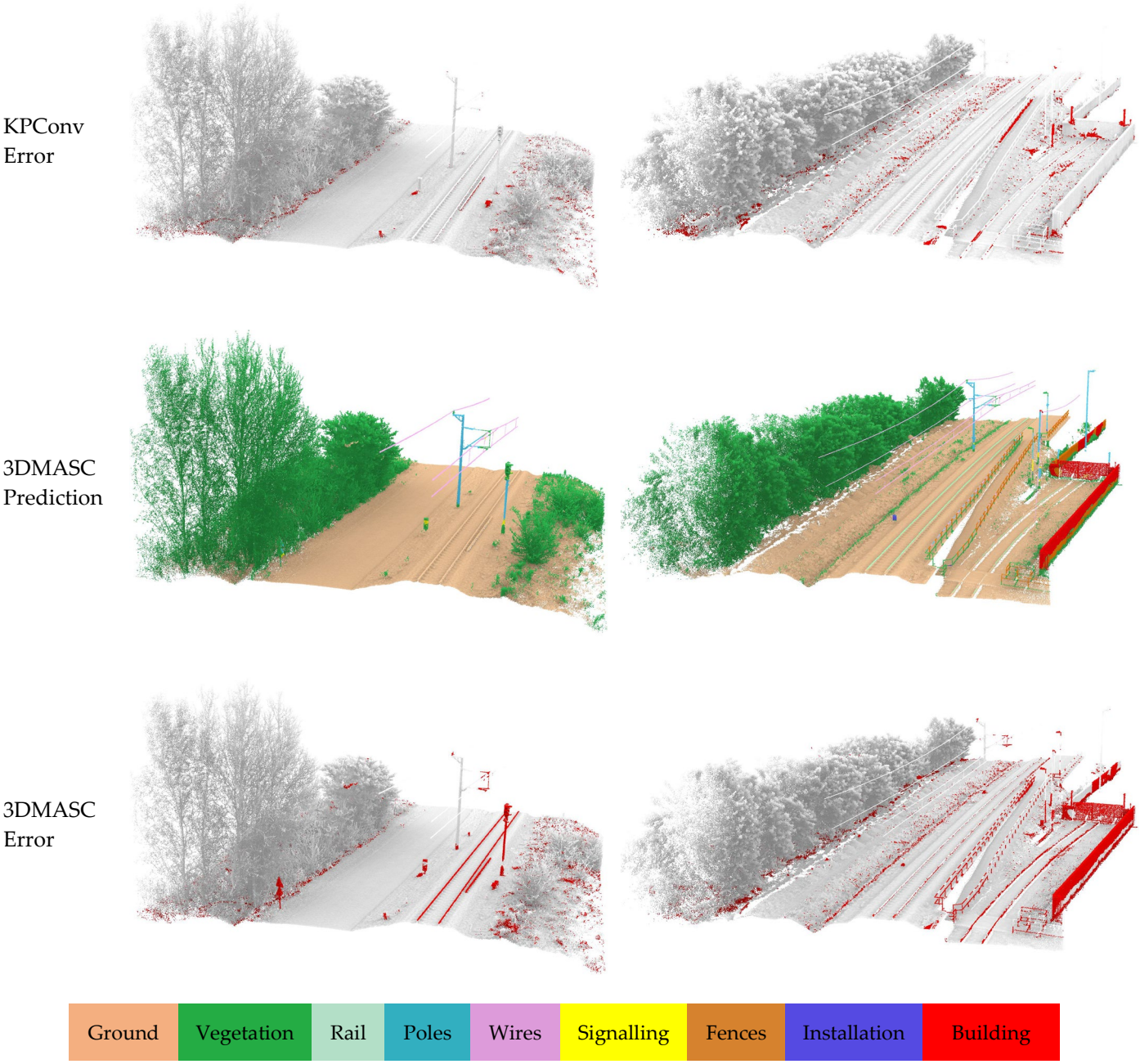
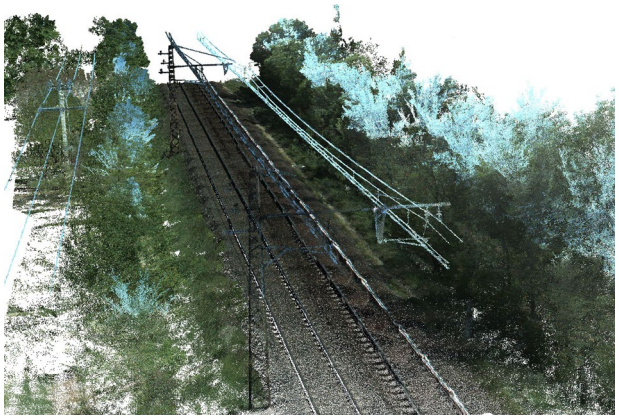
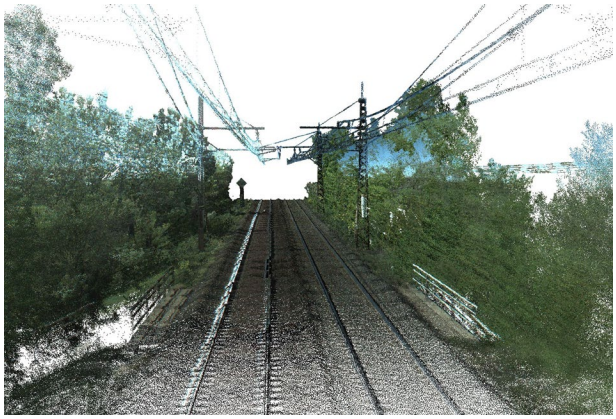


Figure B. Qualitative results on HMLS subset.

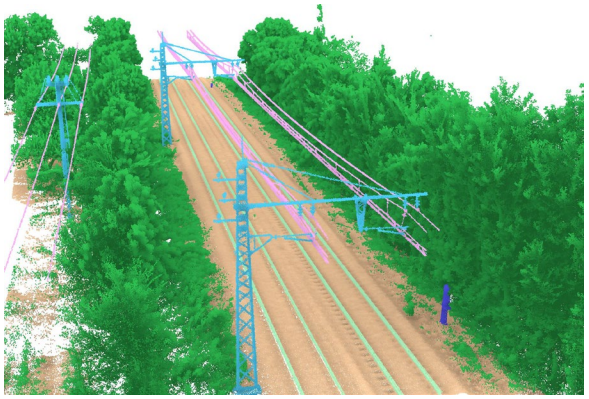
APPENDIX C

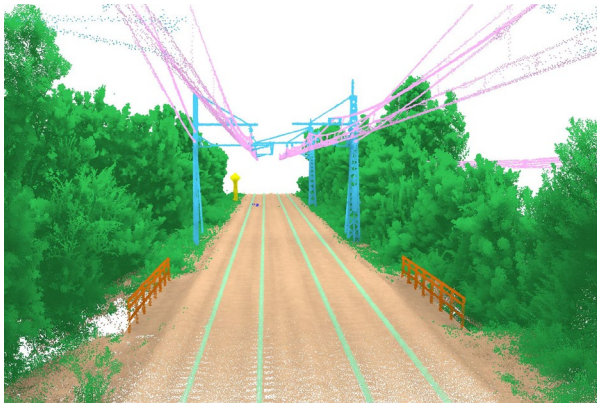
Input



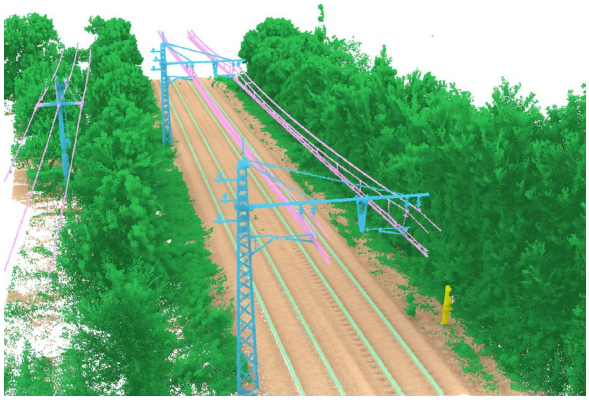


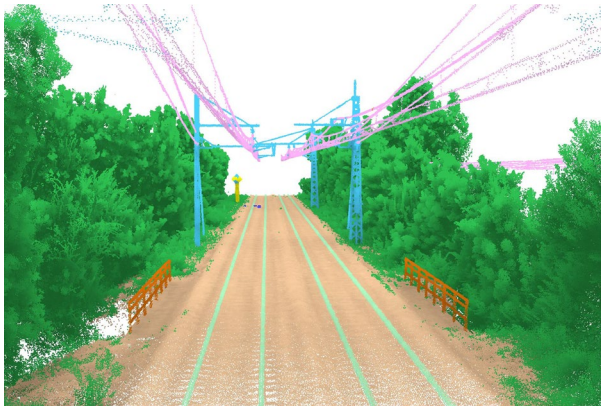
Ground Truth



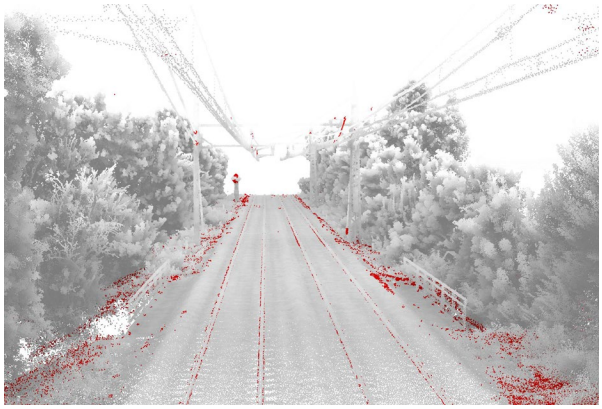


KPConv Prediction





KPConv Error



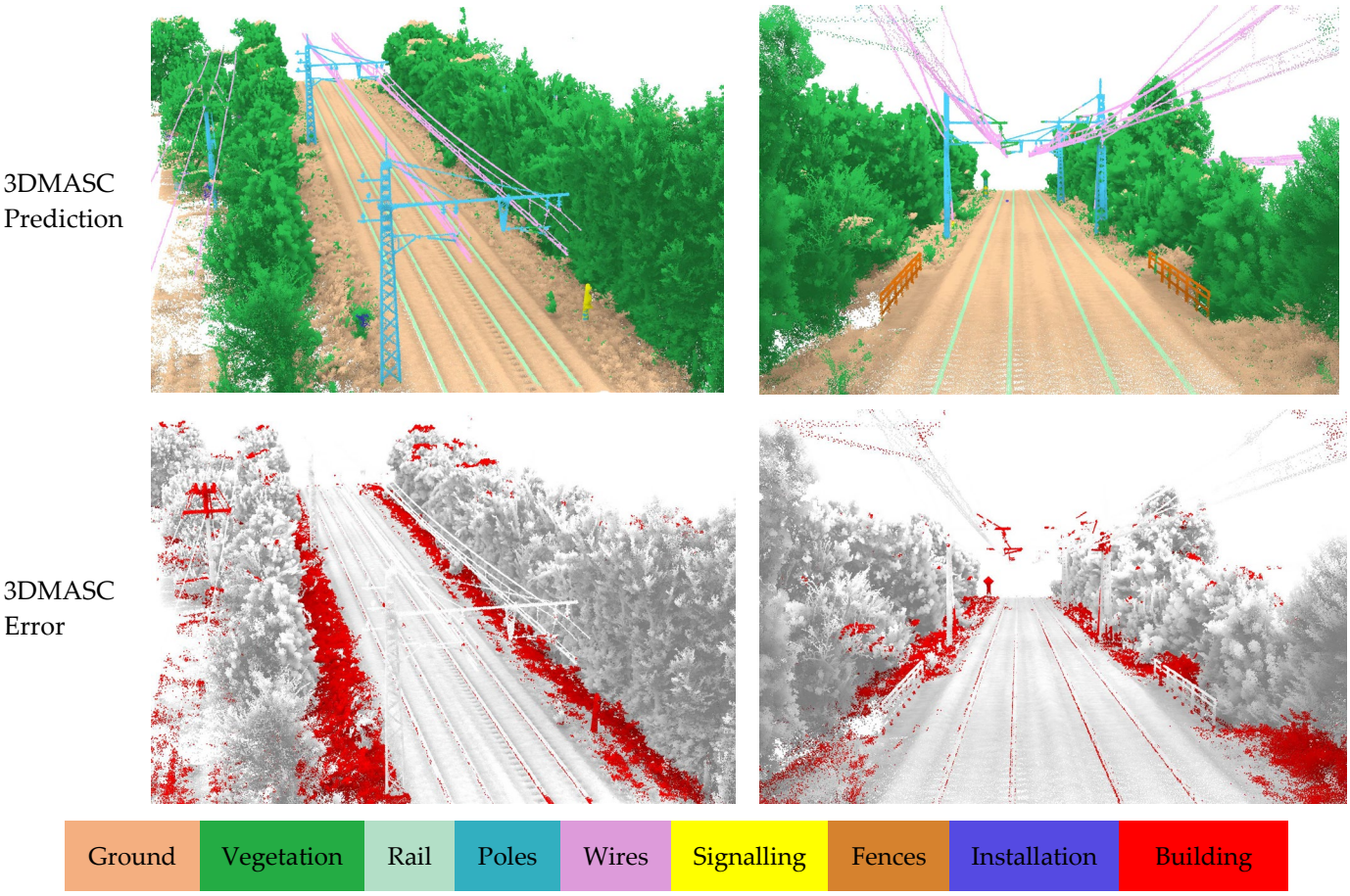
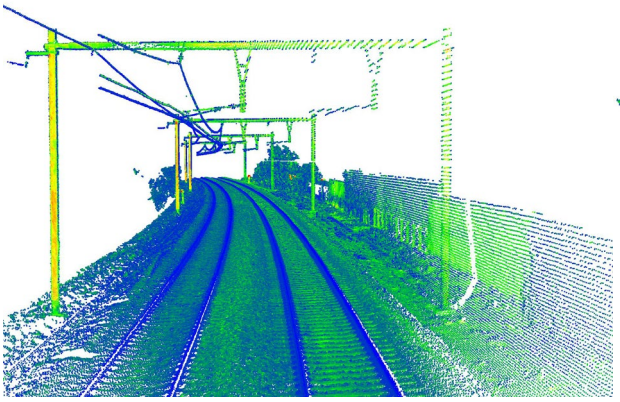
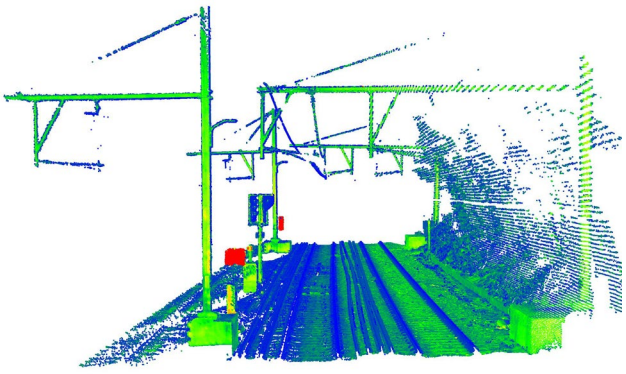


Figure C. Qualitative results on SNCF subset.

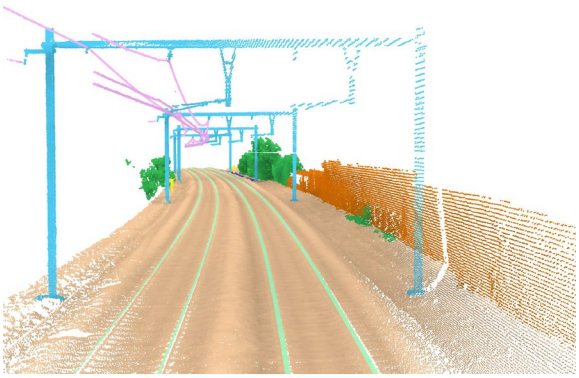
APPENDIX D

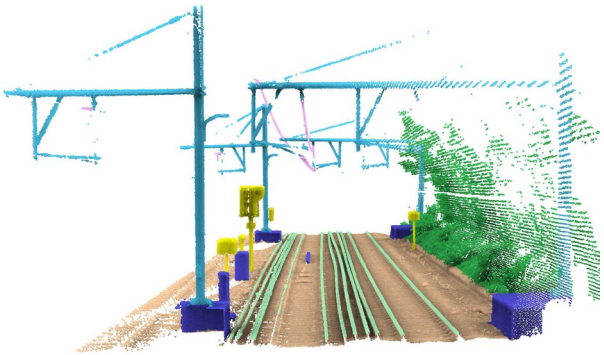
Input



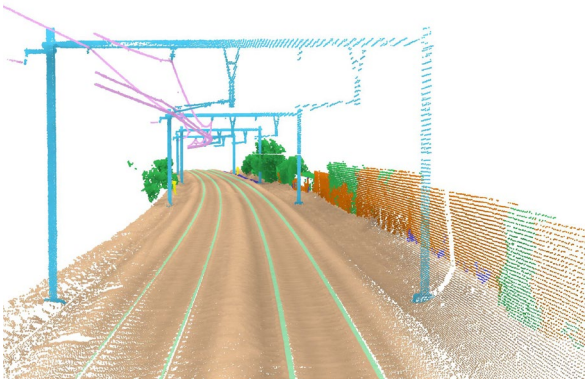


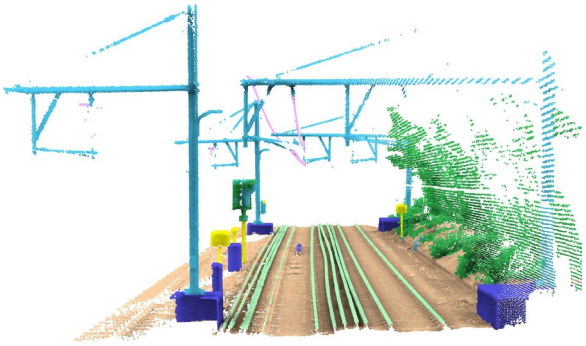
Ground Truth



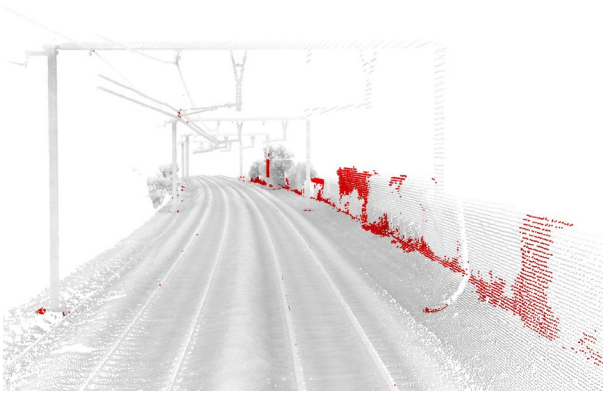


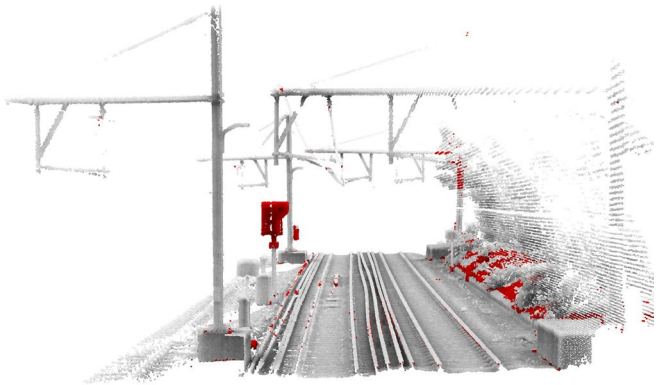
KPConv Prediction





KPConv Error





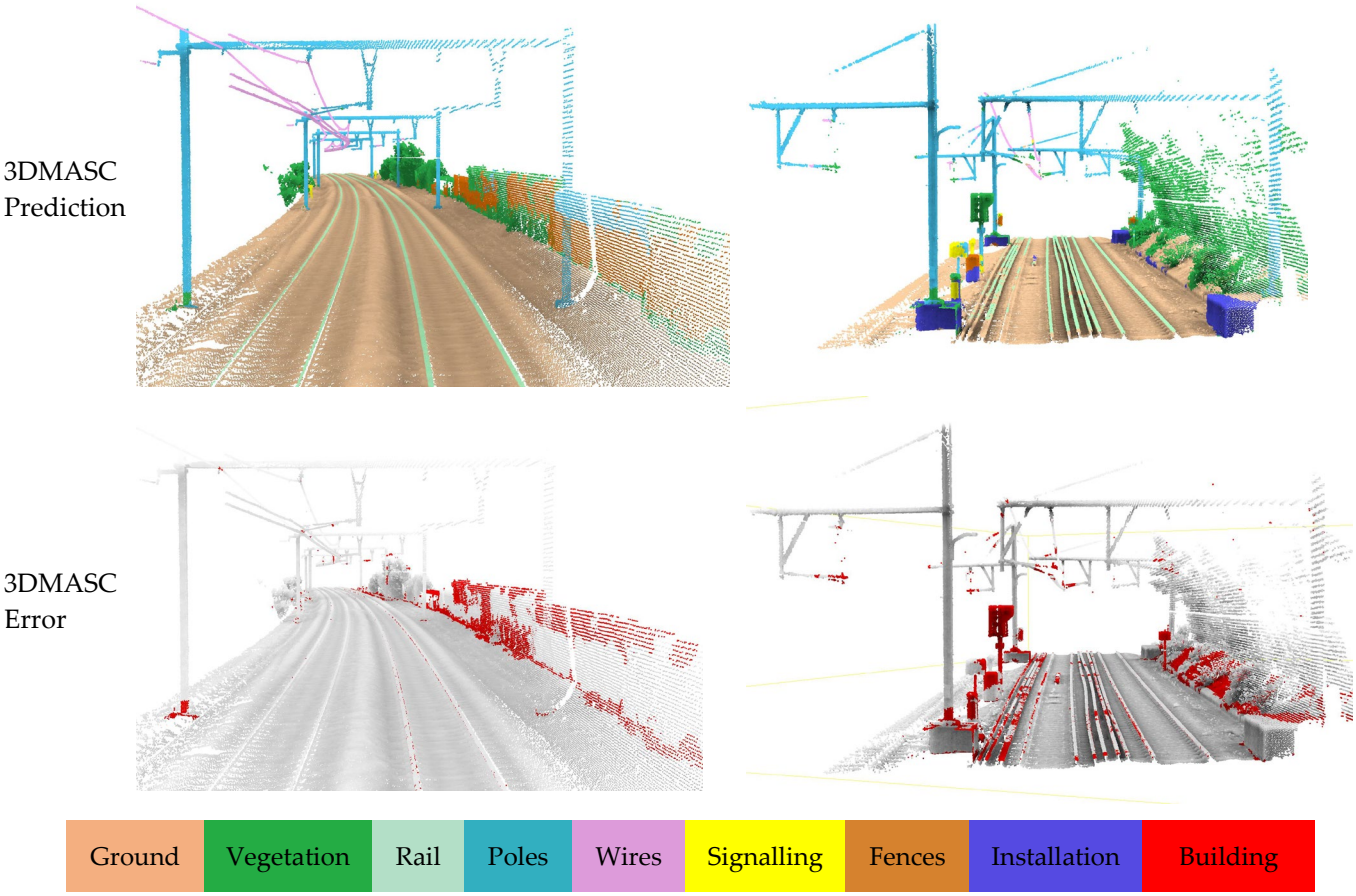


Figure D. Qualitative results on INFRABEL subset.

BIBLIOGRAPHY

[1] M. Li, X. Feng, and Q. Hu, "3D laser point cloud-based geometric digital twin for condition assessment of large diameter pipelines," *Tunnelling and Underground Space Technology*, vol. 142, Dec. 2023, doi: 10.1016/j.tust.2023.105430.

[2] Ö. Zováthi, B. Nagy, and C. Benedek, "Point cloud registration and change detection in urban environment using an onboard Lidar sensor and MLS reference data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 110, Jun. 2022, doi: 10.1016/j.jag.2022.102767.

[3] M. Nafisa and M. Babikir, "Change Detection Techniques using Optical Remote Sensing: A Survey," *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, vol. 17, no. 1, pp. 42–51, 2016, [Online]. Available: http://asrjetsjournal.org/

[4] E. Ghaderpour and T. Vujadinovic, "Change detection within remotely sensed satellite image time series via spectral analysis," *Remote Sens (Basel)*, vol. 12, no. 23, pp. 1–27, Dec. 2020, doi: 10.3390/rs12234001.

[5] Y. Afaq and A. Manocha, "Analysis on change detection techniques for remote sensing applications: A review," *Ecol Inform*, vol. 63, Jul. 2021, doi: 10.1016/j.ecoinf.2021.101310.

[6] T. Betsas, A. Georgopoulos, A. Doulamis, and P. Grussenmeyer, "Deep Learning on 3D Semantic Segmentation: A Detailed Review," Jan. 01, 2025, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/rs17020298.

[7] U. Stilla and Y. Xu, "Change detection of urban objects using 3D point clouds: A review," Mar. 01, 2023, *Elsevier B.V.* doi: 10.1016/j.isprsjprs.2023.01.010.

[8] W. Xiao, H. Cao, M. Tang, Z. Zhang, and N. Chen, "3D urban object change detection from aerial and terrestrial point clouds: A review," Apr. 01, 2023, *Elsevier B.V.* doi: 10.1016/j.jag.2023.103258.

[9] J. P. Virtanen *et al.*, "Nationwide point cloud-The future topographic core data," *ISPRS Int J Geoinf*, vol. 6, no. 8, 2017, doi: 10.3390/ijgi6080243.

[10] J. Théau, "Change Detection," in *Encyclopedia of GIS*, S. Shekhar and H. Xiong, Eds., Boston, MA: Springer US, 2008, pp. 77–84. doi: 10.1007/978-0-387-35973-1_129.

[11] H. Si Salah, S. Ait-Aoudia, A. Rezgui, and S. E. Goldin, "Change detection in urban areas from remote sensing data: a multidimensional classification scheme," *Int J Remote Sens*, vol. 40, no. 17, pp. 6635–6679, 2019, doi: 10.1080/01431161.2019.1583394.

[12] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, "Change detection based on artificial intelligence: State-of-the-art and challenges," *Remote Sens (Basel)*, vol. 12, no. 10, 2020, doi: 10.3390/rs12101688.

[13] J. Seo, W. Park, and T. Kim, “Feature-Based Approach to Change Detection of Small Objects from High-Resolution Satellite Images,” *Remote Sens (Basel)*, vol. 14, no. 3, p. 462, 2022, doi: 10.3390/rs14030462.

[14] T. Han, Y. Tang, X. Yang, Z. Lin, B. Zou, and H. Feng, “Change detection for heterogeneous remote sensing images with improved training of hierarchical extreme learning machine (Helm),” *Remote Sens (Basel)*, vol. 13, no. 23, 2021, doi: 10.3390/rs13234918.

[15] Y. You, J. Cao, and W. Zhou, “A survey of change detection methods based on remote sensing images for multi-source and multi-objective scenarios,” *Remote Sens (Basel)*, vol. 12, no. 15, 2020, doi: 10.3390/RS12152460.

[16] E. Ghaderpour and T. Vujadinovic, “Change detection within remotely sensed satellite image time series via spectral analysis,” *Remote Sens (Basel)*, vol. 12, no. 23, pp. 1–27, 2020, doi: 10.3390/rs12234001.

[17] Y. Afaq and A. Manocha, “Analysis on change detection techniques for remote sensing applications: A review,” *Ecol Inform*, vol. 63, no. May, p. 101310, 2021, doi: 10.1016/j.ecoinf.2021.101310.

[18] A. Goswami *et al.*, “Change Detection in Remote Sensing Image Data Comparing Algebraic and Machine Learning Methods,” *Electronics (Switzerland)*, vol. 11, no. 3, pp. 1–26, 2022, doi: 10.3390/electronics11030431.

[19] A. Asokan and J. Anitha, “Change detection techniques for remote sensing applications: a survey,” *Earth Sci Inform*, pp. 143–160, 2019, doi: 10.1007/s12145-019-00380-5.

[20] B. R. Kiran *et al.*, “Real-time dynamic object detection for autonomous driving using prior 3D-maps,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11133 LNCS, pp. 567–582, 2019, doi: 10.1007/978-3-030-11021-5_35.

[21] E. Khatab, A. Onsy, M. Varley, and A. Abouelfarag, “Vulnerable objects detection for autonomous driving: A review,” *Integration*, vol. 78, no. January, pp. 36–48, 2021, doi: 10.1016/j.vlsi.2021.01.002.

[22] R. Qin, J. Tian, and P. Reinartz, “3D change detection-approaches and applications.”

[23] W. Shuai, F. Jiang, H. Zheng, and J. Li, “MSGATN: A Superpixel-Based Multi-Scale Siamese Graph Attention Network for Change Detection in Remote Sensing Images,” *Applied Sciences (Switzerland)*, vol. 12, no. 10, 2022, doi: 10.3390/app12105158.

[24] Z. Y. Lv, T. F. Liu, J. A. Benediktsson, T. Lei, and Y. L. Wan, “Multi-scale object histogram distance for LCCD using bi-temporal very-high-resolution remote sensing images,” *Remote Sens (Basel)*, vol. 10, no. 11, 2018, doi: 10.3390/rs10111809.

[25] I. de Gélis, S. Lefèvre, and T. Corpetti, “Change detection in urban point clouds: An experimental comparison with simulated 3d datasets,” *Remote Sens (Basel)*, vol. 13, no. 13, pp. 1–29, 2021, doi: 10.3390/rs13132629.

[26] K. Zięba-Kulawik, K. Skoczylas, P. Wężyk, J. Teller, A. Mustafa, and H. Omrani, “Monitoring of urban forests using 3D spatial indices based on LiDAR point clouds and voxel approach,” *Urban For Urban Green*, vol. 65, 2021, doi: 10.1016/j.ufug.2021.127324.

[27] I. De Gelis, S. Lefevre, T. Corpetti, T. Ristorcelli, C. Thenoz, and P. Lassalle, “Benchmarking Change Detection in Urban 3D Point Clouds,” pp. 3352–3355, 2021, doi: 10.1109/igarss47720.2021.9553018.

[28] P. Tompalski *et al.*, “Estimating Changes in Forest Attributes and Enhancing Growth Projections: a Review of Existing Approaches and Future Directions Using Airborne 3D Point Cloud Data,” *Current Forestry Reports*, vol. 7, no. 1, pp. 25–30, 2021, doi: 10.1007/s40725-021-00139-6.

[29] L. Duncanson and R. Dubayah, “Monitoring individual tree-based change with airborne lidar,” *Ecol Evol*, vol. 8, no. 10, pp. 5079–5089, 2018, doi: 10.1002/ece3.4075.

[30] T. Yrttimaa *et al.*, “Structural changes in Boreal forests can be quantified using terrestrial laser scanning,” *Remote Sens (Basel)*, vol. 12, no. 17, pp. 1–20, 2020, doi: 10.3390/RS12172672.

[31] V. Gstaiger, J. Tian, R. Kiefl, and F. Kurz, “2D vs. 3D change detection using aerial imagery to support crisis management of large-scale events,” *Remote Sens (Basel)*, vol. 10, no. 12, pp. 1–19, 2018, doi: 10.3390/rs10122054.

[32] M. Koeva, S. Nikoohemat, S. O. Elberink, J. Morales, C. Lemmen, and J. Zevenbergen, “Towards 3D indoor cadastre based on change detection from point clouds,” *Remote Sens (Basel)*, vol. 11, no. 17, 2019, doi: 10.3390/rs11171972.

[33] M. Awrangjeb, S. A. N. Gilani, and F. U. Siddiqui, “An effective data-driven method for 3-D building roof reconstruction and robust change detection,” *Remote Sens (Basel)*, vol. 10, no. 10, pp. 1–31, 2018, doi: 10.3390/rs10101512.

[34] M. Awrangjeb, “Effective generation and update of a building map database through automatic building change detection from LiDAR point cloud data,” *Remote Sens (Basel)*, vol. 7, no. 10, pp. 14119–14150, 2015, doi: 10.3390/rs71014119.

[35] A. Mayr, M. Rutzinger, and C. Geitner, “Object-based point cloud analysis for landslide and erosion monitoring,” *Photogramm Eng Remote Sensing*, vol. 85, no. 6, pp. 455–462, 2019, doi: 10.14358/PERS.85.6.455.

[36] R. A. Kromer *et al.*, “Automated terrestrial laser scanning with near-real-time change detection - Monitoring of the Séchilienne landslide,” *Earth Surface Dynamics*, vol. 5, no. 2, pp. 293–310, 2017, doi: 10.5194/esurf-5-293-2017.

[37] Z. Bessin *et al.*, "CLIFF CHANGE DETECTION USING SIAMESE KPCONV DEEP NETWORK ON 3D POINT CLOUDS," vol. V, no. June, pp. 6–11, 2022.

[38] Z. Gojcic, L. Schmid, and A. Wieser, "Dense 3D displacement vector fields for point cloud-based landslide monitoring," *Landslides*, vol. 18, no. 12, pp. 3821–3832, 2021, doi: 10.1007/s10346-021-01761-y.

[39] A. Mayr, M. Rutzinger, M. Bremer, S. Oude Elberink, F. Stumpf, and C. Geitner, "Object-based classification of terrestrial laser scanning point clouds for landslide monitoring," *Photogrammetric Record*, vol. 32, no. 160, pp. 377–397, 2017, doi: 10.1111/phor.12215.

[40] T. Meyer, A. Brunn, and U. Stilla, "Automation in Construction Change detection for indoor construction progress monitoring based on BIM , point clouds and uncertainties," *Autom Constr*, vol. 141, no. June, p. 104442, 2022, doi: 10.1016/j.autcon.2022.104442.

[41] U. Okyay, J. Telling, C. L. Glennie, and W. E. Dietrich, "Airborne lidar change detection: An overview of Earth sciences applications," *Earth Sci Rev*, vol. 198, no. June, p. 102929, 2019, doi: 10.1016/j.earscirev.2019.102929.

[42] K. Anders *et al.*, "Multitemporal terrestrial laser scanning point clouds for thaw subsidence observation at Arctic permafrost monitoring sites," *Earth Surf Process Landf*, vol. 45, no. 7, pp. 1589–1600, 2020, doi: 10.1002/esp.4833.

[43] D. Girardeau-Montaut, M. Roux, R. Marc, and G. Thibault, "Change detection on point cloud data acquired with a ground laser scanner," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 36, 2005.

[44] C. Dai, Z. Zhang, and D. Lin, "An object-based bidirectional method for integrated building extraction and change detection between multimodal point clouds," *Remote Sens (Basel)*, vol. 12, no. 10, 2020, doi: 10.3390/rs12101680.

[45] J. Schachtschneider, A. Schlichting, and C. Brenner, "Assessing temporal behavior in lidar point clouds of urban environments," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 42, no. 1W1, pp. 543–550, 2017, doi: 10.5194/isprs-archives-XLII-1-W1-543-2017.

[46] A. K. Aijazi, P. Checchin, and L. Trassoudaine, "Detecting and Updating Changes in Lidar Point Clouds for Automatic 3D Urban Cartography," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 5W2, pp. 7–12, 2013, doi: 10.5194/isprsannals-II-5-W2-7-2013.

[47] J. Gehrung, M. Hebel, M. Arens, and U. Stilla, "A FAST VOXEL-BASED INDICATOR for CHANGE DETECTION USING LOW RESOLUTION OCTREES," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, no. 2/W5, pp. 357–364, 2019, doi: 10.5194/isprs-annals-IV-2-W5-357-2019.

[48] J. Gehrung, M. Hebel, M. Arens, and U. Stilla, "A VOXEL-BASED METADATA STRUCTURE for CHANGE DETECTION in POINT CLOUDS of LARGE-SCALE URBAN AREAS," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, no. 2, pp. 97–104, 2018, doi: 10.5194/isprs-annals-IV-2-97-2018.

[49] A. Harith, L. D. F., C. Dolores, and V. Manuel, "Voxel Change: Big Data–Based Change Detection for Aerial Urban LiDAR of Unequal Densities," *Journal of Surveying Engineering*, vol. 147, no. 4, p. 4021023, Nov. 2021, doi: 10.1061/(ASCE)SU.1943-5428.0000356.

[50] H. P. Moravec, "Sensor Fusion in Certainty Grids for Mobile Robots," *Sensor Devices and Systems for Robotics*, vol. 9, no. 2, pp. 253–276, 1989, doi: 10.1007/978-3-642-74567-6_19.

[51] A. Elfes, "Occupancy Grids: a Probabilistic Framework for Robot Perception and Navigation," Ph.D. thesis, Carnegie Mellon University, 1989.

[52] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," *IEEE Conference on Robotics, Automation and Mechatronics, RAM - Proceedings*, no. November, pp. 225–230, 2013, doi: 10.1109/RAM.2013.6758588.

[53] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors (Switzerland)*, vol. 19, no. 19, pp. 1–22, 2019, doi: 10.3390/s19194188.

[54] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *ArXiv*, pp. 1–27, 2019, doi: 10.1109/tpami.2020.3005434.

[55] Y. Xie, J. Tian, and X. X. Zhu, "Linking Points with Labels in 3D: A Review of Point Cloud Semantic Segmentation," *IEEE Geosci Remote Sens Mag*, vol. 8, no. 4, pp. 38–59, 2020, doi: 10.1109/MGRS.2019.2937630.

[56] B. Ouyang and D. Raviv, "Occlusion guided scene flow estimation on 3D point clouds," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2799–2808, 2021, doi: 10.1109/CVPRW53098.2021.00315.

[57] P. Jund, C. Sweeney, N. Abdo, Z. Chen, and J. Shlens, "Scalable Scene Flow from Point Clouds in the Real World," 2021, [Online]. Available: http://arxiv.org/abs/2103.01306

[58] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE Trans Pattern Anal Mach Intell*, vol. 27, no. 3, pp. 475–480, 2005, doi: 10.1109/TPAMI.2005.63.

[59] S. Wang, Y. Sun, C. Liu, and M. Liu, "PointTrackNet: An End-to-End Network for 3-D Object Detection and Tracking from Point Clouds," *IEEE Robot Autom Lett*, vol. 5, no. 2, pp. 3206–3212, 2020, doi: 10.1109/LRA.2020.2974392.

[60] P. Girão, A. Asvadi, P. Peixoto, and U. Nunes, “3D object tracking in driving environment: A short review and a benchmark dataset,” *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 7–12, 2016, doi: 10.1109/ITSC.2016.7795523.

[61] R. Zeibak and S. Filin, “Change detection via terrestrial laser scanning,” *ISPRS Workshop on Laser Scanning*, no. May 2012, pp. 430–435, 2007.

[62] H. Jiang *et al.*, “A Survey on Deep Learning-Based Change Detection from High-Resolution Remote Sensing Images,” Apr. 01, 2022, *MDPI*. doi: 10.3390/rs14071552.

[63] M. Hemati, M. Hasanlou, M. Mahdianpari, and F. Mohammadimanesh, “A systematic review of landsat data for change detection applications: 50 years of monitoring the earth,” *Remote Sens (Basel)*, vol. 13, no. 15, 2021, doi: 10.3390/rs13152869.

[64] S. Pang, X. Hu, Z. Wang, and Y. Lu, “Object-based analysis of airborne LiDAR data for building change detection,” *Remote Sens (Basel)*, vol. 6, no. 11, pp. 10733–10749, 2014, doi: 10.3390/rs61110733.

[65] S. Xu, G. Vosselman, and S. Oude Elberink, “Detection and Classification of Changes in Buildings from Airborne Laser Scanning Data,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 2, no. 5W2, pp. 343–348, 2013, doi: 10.5194/isprsannals-II-5-W2-343-2013.

[66] J. Chen, J. S. K. Yi, M. Kahoush, E. S. Cho, and Y. K. Cho, “Point cloud scene completion of obstructed building facades with generative adversarial inpainting,” *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–27, 2020, doi: 10.3390/s20185029.

[67] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Niebner, “ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2018, doi: 10.1109/CVPR.2018.00481.

[68] X. Yang *et al.*, “Semantic Segmentation-assisted Scene Completion for LiDAR Point Clouds,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 3555–3562, 2021, doi: 10.1109/IROS51168.2021.9636662.

[69] L. Roldao, R. de Charette, and A. Verroust-Blondet, “3D Semantic Scene Completion: a Survey,” 2021, doi: 10.1007/s11263-021-01504-5.

[70] Y. Zhang, Z. Liu, X. Li, and Y. Zang, “Data-driven point cloud objects completion,” *Sensors (Switzerland)*, vol. 19, no. 7, pp. 1–13, 2019, doi: 10.3390/s19071514.

[71] T. Czerniawski, J. W. Ma, and Fernanda Leite, “Automated building change detection with amodal completion of point clouds,” *Autom Constr*, vol. 124, no. May 2020, p. 103568, 2021, doi: 10.1016/j.autcon.2021.103568.

[72] N. Singer and V. K. Asari, “View-Agnostic Point Cloud Generation for Occlusion Reduction in Aerial Lidar,” 2022.

[73] F. Poux and R. Billen, “Voxel-Based 3D Point Cloud Semantic Segmentation : Unsupervised Geometric and Relationship Featuring vs Deep Learning Methods,” pp. 1–35, 2019, doi: 10.3390/ijgi8050213.

[74] J. Wang, F. Gao, J. Dong, S. Zhang, and Q. Du, “Change Detection From Synthetic Aperture Radar Images via Graph-Based Knowledge Supplement Network,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 15, pp. 1823–1836, 2022, doi: 10.1109/JSTARS.2022.3146167.

[75] L. Khelifi and M. Mignotte, “Deep Learning for Change Detection in Remote Sensing Images: Comprehensive Review and Meta-Analysis,” *IEEE Access*, vol. 8, pp. 126385–126400, 2020, doi: 10.1109/ACCESS.2020.3008036.

[76] A. Shafique, G. Cao, Z. Khan, M. Asad, and M. Aslam, “Deep Learning-Based Change Detection in Remote Sensing Images: A Review,” *Remote Sens (Basel)*, vol. 14, no. 4, pp. 1–40, 2022, doi: 10.3390/rs14040871.

[77] H. Chen, Z. Qi, and Z. Shi, “Remote Sensing Image Change Detection with Transformers,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2022, doi: 10.1109/TGRS.2021.3095166.

[78] L. Shen *et al.*, “S2looking: A satellite side-looking dataset for building change detection,” *Remote Sens (Basel)*, vol. 13, no. 24, pp. 1–23, 2021, doi: 10.3390/rs13245094.

[79] S. Tian, A. Ma, Z. Zheng, and Y. Zhong, “Hi-UCD: A Large-scale Dataset for Urban Semantic Change Detection in Remote Sensing Imagery,” no. 2, pp. 1–6, 2020, [Online]. Available: http://arxiv.org/abs/2011.03247

[80] K. Yang *et al.*, “Semantic Change Detection with Asymmetric Siamese Networks,” pp. 1–14, 2020, [Online]. Available: http://arxiv.org/abs/2010.05687

[81] D. Peng, L. Bruzzone, Y. Zhang, H. Guan, H. DIng, and X. Huang, “SemiCDNet: A Semisupervised Convolutional Neural Network for Change Detection in High Resolution Remote-Sensing Images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 7, pp. 5891–5906, 2021, doi: 10.1109/TGRS.2020.3011913.

[82] H. Chen and Z. Shi, “A spatial-temporal attention-based method and a new dataset for remote sensing image change detection,” *Remote Sens (Basel)*, vol. 12, no. 10, 2020, doi: 10.3390/rs12101662.

[83] R. Caye Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, “HRSCD - High Resolution Semantic Change Detection Dataset,” 2020, *IEEE Dataport*. doi: 10.21227/azv7-ta17.

[84] V. Růžička, S. D’Aronco, J. D. Wegner, and K. Schindler, “Deep active learning in remote sensing for data efficient change detection,” *CEUR Workshop Proc*, vol. 2766, 2020.

[85] Z. Zhu, “Change detection using landsat time series: A review of frequencies, preprocessing, algorithms, and applications,” *ISPRS Journal of Photogrammetry*

and Remote Sensing, vol. 130, pp. 370–384, 2017, doi: 10.1016/j.isprsjprs.2017.06.013.

[86] A. C. Carrilho, M. Galo, and R. C. Dos Santos, "Statistical outlier detection method for airborne LiDAR data," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 42, no. 1, pp. 87–92, 2018, doi: 10.5194/isprs-archives-XLII-1-87-2018.

[87] C. Hu, Z. Pan, and P. Li, "A 3D point cloud filtering method for leaves based on manifold distance and normal estimation," *Remote Sens (Basel)*, vol. 11, no. 2, pp. 1–19, 2019, doi: 10.3390/rs11020198.

[88] R. Gao, M. Li, S. J. Yang, and K. Cho, "Reflective Noise Filtering of Large-Scale Point Cloud Using Transformer," *Remote Sens (Basel)*, vol. 14, no. 3, 2022, doi: 10.3390/rs14030577.

[89] S. Cai, W. Zhang, X. Liang, P. Wan, J. Qi, and S. Yu, "Filtering Airborne LiDAR Data Through Complementary Cloth Simulation and Progressive TIN Densification Filters," 2016.

[90] Y. Li, H. Wu, H. Xu, R. An, J. Xu, and Q. He, "A gradient-constrained morphological filtering algorithm for airborne LiDAR," *Opt Laser Technol*, vol. 54, pp. 288–296, 2013, doi: 10.1016/j.optlastec.2013.06.007.

[91] Q. Chen, P. Gong, D. Baldocchi, and G. Xie, "Filtering airborne laser scanning data with morphological methods," *Photogramm Eng Remote Sensing*, vol. 73, no. 2, pp. 175–185, 2007, doi: 10.14358/PERS.73.2.175.

[92] Z. Hui, Y. Hu, Y. Z. Yevenyo, and X. Yu, "An improved morphological algorithm for filtering airborne LiDAR point cloud based on multi-level kriging interpolation," *Remote Sens (Basel)*, vol. 8, no. 1, pp. 12–16, 2016, doi: 10.3390/rs8010035.

[93] T. J. Pingel, K. C. Clarke, and W. A. McBride, "An improved simple morphological filter for the terrain classification of airborne LIDAR data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 77, pp. 21–30, 2013, doi: 10.1016/j.isprsjprs.2012.12.002.

[94] H. Hu *et al.*, "An adaptive surface filter for airborne laser scanning point clouds by means of regularization and bending energy," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 92, pp. 98–111, 2014, doi: 10.1016/j.isprsjprs.2014.02.014.

[95] Q. Chen, H. Wang, H. Zhang, M. Sun, and X. Liu, "A point cloud filtering approach to generating DTMs for steep mountainous areas and adjacent residential areas," *Remote Sens (Basel)*, vol. 8, no. 1, 2016, doi: 10.3390/rs8010071.

[96] D. Mongus and B. Žalik, "Parameter-free ground filtering of LiDAR data for automatic DTM generation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 67, no. 1, pp. 1–12, 2012, doi: 10.1016/j.isprsjprs.2011.10.002.

[97] G. Vosselman, “Slope based filtering of laser altimetry data,” *International Archives of Photogrammetry and Remote Sensing, Vol. 33, Part B3/2*, vol. 33, no. Part B3/2, pp. 678–684, 2000, doi: 10.1016/S0924-2716(98)00009-4.

[98] X. Meng, N. Currit, and K. Zhao, “Ground filtering algorithms for airborne LiDAR data: A review of critical issues,” *Remote Sens (Basel)*, vol. 2, no. 3, pp. 833–860, 2010, doi: 10.3390/rs2030833.

[99] X. Lin and J. Zhang, “Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments,” *Remote Sens (Basel)*, vol. 6, no. 2, pp. 1294–1326, 2014, doi: 10.3390/rs6021294.

[100] R. Richter, M. Behrens, and J. Döllner, “Object class segmentation of massive 3D point clouds of urban areas using point cloud topology,” *Int J Remote Sens*, vol. 34, no. 23, pp. 8408–8424, 2013, doi: 10.1080/01431161.2013.838710.

[101] M. Yan, T. Blaschke, Y. Liu, and L. Wu, “An object-based analysis filtering algorithm for airborne laser scanning,” *Int J Remote Sens*, vol. 33, no. 22, pp. 7099–7116, 2012, doi: 10.1080/01431161.2012.699694.

[102] G. Sithole and G. Vosselman, “FILTERING OF AIRBORNE LASER SCANNER DATA BASED ON SEGMENTED POINT CLOUDS,” *GeoInformation Science*, vol. 36, no. part 3, pp. 66–71, 2005, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.136.1111&rep=rep1&type=pdf

[103] B. Bellekens, V. Spruyt, and M. Weyn, “A Survey of Rigid 3D Pointcloud Registration Algorithms,” *AMBIENT 2014, The Fourth International Conference on Ambient Computing, Applications, Services and Technologies. 2014.*, no. c, pp. 8–13, 2014.

[104] Z. Zhang, Y. Dai, and J. Sun, “Deep learning based point cloud registration: an overview,” *Virtual Reality and Intelligent Hardware*, vol. 2, no. 3, pp. 222–246, 2020, doi: 10.1016/j.vrih.2020.05.002.

[105] X. Huang, G. Mei, J. Zhang, and R. Abbas, “A comprehensive survey on point cloud registration,” pp. 1–17, 2021, [Online]. Available: http://arxiv.org/abs/2103.02690

[106] P. Axelsson, “DEM Generation from Laser Scanner Data Using Adaptive TIN Models,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 33, pp. 110–117, 2000.

[107] L. Matikainen, J. Hyyppä, and H. Kaartinen, “Automatic detection of changes from laser scanner and aerial image data for updating building maps,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 35, 2004.

[108] T. T. Vu, M. Matsuoka, and F. Yamazaki, “LIDAR-based change detection of buildings in dense urban areas,” *International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 5, no. October, pp. 3413–3416, 2004, doi: 10.1109/igarss.2004.1370438.

[109] G. Vosselman, G. Sithole, and B. G. H. Gorte, "Change Detection for Updating Medium Scale Maps Using Laser Altimetry," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 34, no. Part B3, p. 6, 2004.

[110] K. Choi, I. Lee, and S. Kim, "A Feature Based Approach to Automatic Change Detection from Lidar Data in Urban Areas," *International Society for Photogrammetry and Remote Sensing Vol. XXXVIII, Part 3/W8*, vol. XXXVIII, no. C, p. 6, 2009.

[111] L. Matikainen, J. Hyyppä, E. Ahokas, L. Markelin, and H. Kaartinen, "Automatic detection of buildings and changes in buildings for updating of maps," *Remote Sens (Basel)*, vol. 2, no. 5, pp. 1217–1248, 2010, doi: 10.3390/rs2051217.

[112] C. Stal, F. Tack, P. de Maeyer, A. de Wulf, and R. Goossens, "Airborne photogrammetry and lidar for DSM extraction and 3D change detection over an urban area - a comparative study," *Int J Remote Sens*, vol. 34, no. 4, pp. 1087–1110, 2013, doi: 10.1080/01431161.2012.717183.

[113] J. A. Malpica, M. C. Alonso, F. Papí, A. Arozarena, and A. M. De Agirrea, "Change detection of buildings from satellite imagery and lidar data," *Int J Remote Sens*, vol. 34, no. 5, pp. 1652–1675, 2013, doi: 10.1080/01431161.2012.725483.

[114] T. A. Teo and T. Y. Shih, "Lidar-based change detection and change-type determination in urban areas," *Int J Remote Sens*, vol. 34, no. 3, pp. 968–981, 2013, doi: 10.1080/01431161.2012.714504.

[115] X. Zhang and C. Glennie, "Change detection from differential airborne LiDAR using a weighted anisotropic iterative closest point algorithm," in *2014 IEEE Geoscience and Remote Sensing Symposium*, 2014, pp. 2162–2165. doi: 10.1109/IGARSS.2014.6946895.

[116] F. Tang, Z. Xiang, D. Teng, B. Hu, and Y. Bai, "A multilevel change detection method for buildings using laser scanning data and GIS data," in *2015 IEEE International Conference on Digital Signal Processing (DSP)*, 2015, pp. 1011–1015. doi: 10.1109/ICDSP.2015.7252030.

[117] S. Xu, G. Vosselman, and S. O. Elberink, "Detection and classification of changes in buildings from airborne laser scanning data," *Remote Sens (Basel)*, vol. 7, no. 12, pp. 17051–17076, 2015, doi: 10.3390/rs71215867.

[118] H. Xu, L. Cheng, M. Li, Y. Chen, and L. Zhong, "Using octrees to detect changes to buildings and trees in the urban environment from airborne liDAR data," *Remote Sens (Basel)*, vol. 7, no. 8, pp. 9682–9704, 2015, doi: 10.3390/rs70809682.

[119] S. Du *et al.*, "Building change detection using old aerial images and new LiDAR data," *Remote Sens (Basel)*, vol. 8, no. 12, pp. 1–22, 2016, doi: 10.3390/rs8121030.

[120] L. Matikainen, J. Hyyppä, and P. Litkey, "Multispectral airborne laser scanning for automated map updating," in *International Archives of the Photogrammetry,*

Remote Sensing and Spatial Information Sciences - ISPRS Archives, 2016, pp. 323–330. doi: 10.5194/isprsarchives-XLI-B3-323-2016.

[121] L. Matikainen, K. Karila, J. Hyyppä, P. Litkey, E. Puttonen, and E. Ahokas, "Object-based analysis of multispectral airborne laser scanner data for land cover classification and map updating," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 128, pp. 298–313, 2017, doi: 10.1016/j.isprsjprs.2017.04.005.

[122] K. Zhao, J. C. Suarez, M. Garcia, T. Hu, C. Wang, and A. Londo, "Utility of multitemporal lidar for forest and carbon monitoring: Tree growth, biomass dynamics, and carbon flux," *Remote Sens Environ*, vol. 204, no. September 2016, pp. 883–897, 2018, doi: 10.1016/j.rse.2017.09.007.

[123] D. Marinelli, C. Paris, and L. Bruzzone, "A Novel Approach to 3-D Change Detection in Multitemporal LiDAR Data Acquired in Forest Areas," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 6, pp. 3030–3046, 2018, doi: 10.1109/TGRS.2018.2789660.

[124] Z. Zhang, G. Vosselman, M. Gerke, C. Persello, D. Tuia, and M. Y. Yang, "CHANGE DETECTION between DIGITAL SURFACE MODELS from AIRBORNE LASER SCANNING and DENSE IMAGE MATCHING USING CONVOLUTIONAL NEURAL NETWORKS," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, no. 2/W5, pp. 453–460, 2019, doi: 10.5194/isprs-annals-IV-2-W5-453-2019.

[125] Z. Zhang, G. Vosselman, M. Gerke, C. Persello, D. Tuia, and M. Y. Yang, "Detecting building changes between airborne laser scanning and photogrammetric data," *Remote Sens (Basel)*, vol. 11, no. 20, 2019, doi: 10.3390/rs11202417.

[126] A. Fekete and M. Cserep, "Tree segmentation and change detection of large urban areas based on airborne LiDAR," *Comput Geosci*, vol. 156, no. June, p. 104900, 2021, doi: 10.1016/j.cageo.2021.104900.

[127] R. Huang, Y. Xu, L. Hoegner, and U. Stilla, "Semantics-aided 3D change detection on construction sites using UAV-based photogrammetric point clouds," *Autom Constr*, no. November, p. 104057, 2021, doi: 10.1016/j.autcon.2021.104057.

[128] T. Ku *et al.*, *SHREC 2021: 3D point cloud change detection for street scenes*, vol. 99. Elsevier Ltd, 2021. doi: 10.1016/j.cag.2021.07.004.

[129] I. De Gelis *et al.*, "Détection de changements urbains 3D par un réseau Siamois sur nuage de points," 2021.

[130] T. H. G. Tran, C. Ressl, and N. Pfeifer, "Integrated change detection and classification in urban areas based on airborne laser scanning point clouds," *Sensors (Switzerland)*, vol. 18, no. 2, 2018, doi: 10.3390/s18020448.

[131] Z. Zhang, "Photogrammetric point clouds: quality assessment, filtering, and change detection," Faculty of Geo-Information Science and Earth Observation, University of Twente, 2022. doi: 10.3990/1.9789036552653.

[132] I. de Gélis, S. Lefèvre, and T. Corpetti, “DC3DCD: unsupervised learning for multiclass 3D point cloud change detection,” May 2023, doi: 10.1016/j.isprsjprs.2023.10.022.

[133] W. Zhan, R. Cheng, and J. Chen, “PGN3DCD: Prior-Knowledge-Guided Network for Urban 3D Point Cloud Change Detection,” *IEEE Transactions on Geoscience and Remote Sensing*, 2024, doi: 10.1109/TGRS.2024.3436854.

[134] R. D. Williams, “DEMs of Difference,” *Geomorphological Techniques*, vol. 2, no. January 2012, pp. 1–17, 2012.

[135] G. Bailey, Y. Li, N. McKinney, D. Yoder, W. Wright, and R. Washington-Allen, “Las2DoD: Change Detection Based on Digital Elevation Models Derived from Dense Point Clouds with Spatially Varied Uncertainty,” *Remote Sens (Basel)*, vol. 14, no. 7, Apr. 2022, doi: 10.3390/rs14071537.

[136] C. P. Scott *et al.*, “Statewide USGS 3DEP Lidar Topographic Differencing Applied to Indiana, USA,” *Remote Sens (Basel)*, vol. 14, no. 4, pp. 1–19, 2022, doi: 10.3390/rs14040847.

[137] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: Measuring Error on Simplified Surfaces,” *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998, doi: 10.1111/1467-8659.00236.

[138] T. B. Barnhart and B. T. Crosby, “Comparing two methods of surface change detection on an evolving thermokarst using high-temporal-frequency terrestrial laser scanning, Selawik River, Alaska,” *Remote Sens (Basel)*, vol. 5, no. 6, pp. 2813–2837, 2013, doi: 10.3390/rs5062813.

[139] D. Lague, N. Brodu, and J. Leroux, “Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N-Z),” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 82, pp. 10–26, 2013, doi: 10.1016/j.isprsjprs.2013.04.009.

[140] V. Zahs, L. Winiwarter, K. Anders, J. G. Williams, M. Rutzinger, and B. Höfle, “Correspondence-driven plane-based M3C2 for lower uncertainty in 3D topographic change quantification,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 183, no. April 2021, pp. 541–559, 2022, doi: 10.1016/j.isprsjprs.2021.11.018.

[141] A. Wagner, “A new approach for geo-monitoring using modern total stations and RGB + D images,” *Measurement (Lond)*, vol. 82, pp. 64–74, 2016, doi: 10.1016/j.measurement.2015.12.025.

[142] Z. Gojcic, C. Zhou, and A. Wieser, “F2S3: Robustified determination of 3D displacement vector fields using deep learning,” *Journal of Applied Geodesy*, vol. 14, no. 2, pp. 177–189, 2020, doi: 10.1515/jag-2019-0044.

[143] Z. Gojcic, C. Zhou, and A. Wieser, “Robust point correspondences for point cloud based deformation monitoring of natural structures,” *4th Joint International Symposium on Deformation Monitoring (JISDM)*, no. May, pp. 15–17, 2019.

[144] F. Poiesi and D. Boscaini, “Distinctive 3D local deep descriptors,” *Proceedings - International Conference on Pattern Recognition*, pp. 5720–5727, 2020, doi: 10.1109/ICPR48806.2021.9411978.

[145] H. Thomas, C. R. Qi, J. E. Deschaud, B. Marcotegui, F. Goulette, and L. Guibas, “KPConv: Flexible and deformable convolution for point clouds,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-Octob, pp. 6410–6419, 2019, doi: 10.1109/ICCV.2019.00651.

[146] S. Krishnan *et al.*, “Opentopography: A Services Oriented Architecture for community access to LIDAR topography,” in *COM.Geo 2011 - 2nd International Conference on Computing for Geospatial Research and Applications*, in ACM International Conference Proceeding Series. 2011. doi: 10.1145/1999320.1999327.

[147] C. J. Crosby, R. J. Arrowsmith, V. Nandigam, and C. Baru, “Online access and processing of LiDAR topography data,” in *Geoinformatics*, G. R. Keller and C. Baru, Eds., Cambridge University Press, 2011, p. 251265. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511976308.017

[148] C. Scott, M. Phan, V. Nandigam, C. Crosby, and J. R. Arrowsmith, “Measuring change at Earth’s surface: On-demand vertical and three-dimensional topographic differencing implemented in OpenTopography,” *Geosphere*, vol. 17, no. 4, pp. 1318–1332, 2021, doi: 10.1130/GES02259.1.

[149] B. Cemellini, W. van Opstal, C.-K. Wang, and D. Xenakis, “Chronocity: Towards an Open Point Cloud Map supporting on-the-fly change detection,” 2017.

[150] M. Hebel, M. Arens, and U. Stilla, “Change detection in urban areas by object-based analysis and on-the-fly comparison of multi-view ALS data,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 86, pp. 52–64, 2013, doi: 10.1016/j.isprsjprs.2013.09.005.

[151] S. Vos, R. Lindenbergh, and S. de Vries, “CoastScan: Continuous Monitoring of Coastal Change Using Terrestrial Laser Scanning,” *Coastal Dynamics*, vol. 2, no. 233, pp. 1518–1528, 2017.

[152] K. Anders, L. Winiwarter, H. Mara, R. Lindenbergh, S. E. Vos, and B. Höfle, “Fully automatic spatiotemporal segmentation of 3D LiDAR time series for the extraction of natural surface changes,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 173, no. November 2020, pp. 297–308, 2021, doi: 10.1016/j.isprsjprs.2021.01.015.

[153] M. Fehr *et al.*, “TSDF-based change detection for consistent long-term dense reconstruction and dynamic object discovery,” *Proc IEEE Int Conf Robot Autom*, pp. 5237–5244, 2017, doi: 10.1109/ICRA.2017.7989614.

[154] S. Vos *et al.*, “A high-resolution 4D terrestrial laser scan dataset of the Kijkduin beach-dune system, The Netherlands,” *Sci Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1038/s41597-022-01291-9.

[155] K. Anders, L. Winiwarter, and B. Hofle, “Improving Change Analysis from Near-Continuous 3D Time Series by Considering Full Temporal Information,” *IEEE Geoscience and Remote Sensing Letters*, vol. 19, 2022, doi: 10.1109/LGRS.2022.3148920.

[156] J. Gehrung, M. Hebel, M. Arens, and U. Stilla, “AN APPROACH to EXTRACT MOVING OBJECTS from MLS DATA USING A VOLUMETRIC BACKGROUND REPRESENTATION,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, no. 1W1, pp. 107–114, 2017, doi: 10.5194/isprs-annals-IV-1-W1-107-2017.

[157] J. Zhu *et al.*, “TUM-MLS-2016: An annotated mobile LiDAR dataset of the TUM city campus for semantic point cloud interpretation in urban areas,” *Remote Sens (Basel)*, vol. 12, no. 11, 2020, doi: 10.3390/rs12111875.

[158] G. Iris, S. Lefèvre, and T. Corpetti, “Urban Point Clouds Simulated Dataset for 3D Change Detection.” [Online]. Available: https://geo.data.gouv.

[159] E. Palazzolo, “Fast Image-Based Geometric Change Detection in a 3D Model,” no. ii, pp. 6308–6315, 2017.

[160] C. J. Crosby, R. J. Arrowsmith, V. Nandigam, and C. Baru, “Online access and processing of LiDAR topography data,” in *Geoinformatics*, G. R. Keller and C. Baru, Eds., Cambridge University Press, 2011, p. 251{\textendash}265. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511976308.017

[161] V. Zahs *et al.*, “Correspondence-driven plane-based M3C2 for quantification of 3D topographic change with lower uncertainty [Data and Source Code],” 2021, *heiDATA*. doi: 10.11588/data/TGSVUI.

[162] L. Winiwarter, K. Anders, and B. Höfle, “M3C2-EP: Pushing the limits of 3D topographic point cloud change detection by error propagation,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 178, no. October 2020, pp. 240–258, 2021, doi: 10.1016/j.isprsjprs.2021.06.011.

[163] A. Mayr, M. Bremer, M. Rutzinger, and P. C. Quality, “3D POINT ERRORS AND CHANGE DETECTION ACCURACY OF UNMANNED AERIAL VEHICLE LASER SCANNING DATA,” vol. V, pp. 765–772, 2020.

[164] L. Winiwarter, K. Anders, D. Wujanz, and B. Höfle, “Influence of Ranging Uncertainty of Terrestrial Laser Scanning on Change Detection in Topographic 3D Point Clouds,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, no. 2, pp. 789–796, 2020, doi: 10.5194/isprs-annals-V-2-2020-789-2020.

[165] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A Survey on Deep Transfer Learning,” Aug. 2018, [Online]. Available: http://arxiv.org/abs/1808.01974

[166] Y. Chen *et al.*, “PointMixup: Augmentation for Point Clouds,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and*

Lecture Notes in Bioinformatics), vol. 12348 LNCS, pp. 330–345, 2020, doi: 10.1007/978-3-030-58580-8_20.

[167] P. Šebek, Š. Pokorný, P. Vacek, and T. Svoboda, "Real3D-Aug: Point Cloud Augmentation by Placing Real Objects with Occlusion Handling for 3D Detection and Segmentation," pp. 1–7, 2022, [Online]. Available: http://arxiv.org/abs/2206.07634

[168] C. L. Li, M. Zaheer, Y. Zhang, B. Póczos, and R. Salakhutdinov, "Point cloud gan," *Deep Generative Models for Highly Structured Data, DGS@ICLR 2019 Workshop*, 2019.

[169] B. Nagy, L. Kovacs, and C. Benedek, "ChangeGAN: A Deep Network for Change Detection in Coarsely Registered Point Clouds," *IEEE Robot Autom Lett*, vol. 6, no. 4, pp. 8277–8284, 2021, doi: 10.1109/LRA.2021.3105721.

[170] W. Yuan *et al.*, "Graph neural network based multi-feature fusion for building change detection," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 43, no. B3-2021, pp. 377–382, 2021, doi: 10.5194/isprs-archives-XLIII-B3-2021-377-2021.

[171] J. Wu *et al.*, "A multiscale graph convolutional network for change detection in homogeneous and heterogeneous remote sensing images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 105, p. 102615, 2021, doi: 10.1016/j.jag.2021.102615.

[172] H. Kim, S. B. Lee, and K. S. Min, "Shoreline change analysis using airborne LiDAR bathymetry for coastal monitoring," in *Journal of Coastal Research*, Coastal Education Research Foundation Inc., Mar. 2017, pp. 269–273. doi: 10.2112/SI79-055.1.

[173] L. Zhang *et al.*, "Object-based 3D building change detection using point-level change indicators," Apr. 01, 2023, *Elsevier B.V.* doi: 10.1016/j.jag.2023.103293.

[174] D. Robert, H. Raguet, and L. Landrieu, "Efficient 3D Semantic Segmentation with Superpoint Transformer," Jun. 2023, [Online]. Available: http://arxiv.org/abs/2306.08045

[175] L. Tan, X. Zuo, and X. Cheng, "CGMNet: Semantic Change Detection via a Change-Aware Guided Multi-Task Network," *Remote Sens (Basel)*, vol. 16, no. 13, Jul. 2024, doi: 10.3390/rs16132436.

[176] Y. Zhu, L. Li, K. Chen, C. Liu, F. Zhou, and Z. Shi, "Semantic-CD: Remote Sensing Image Semantic Change Detection towards Open-vocabulary Setting," Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.06808

[177] Y. Gao *et al.*, "SHREC 2023: Point cloud change detection for city scenes," *Computers and Graphics (Pergamon)*, vol. 115, pp. 35–42, Oct. 2023, doi: 10.1016/j.cag.2023.06.025.

[178] Z. Wang, Y. Zhang, L. Luo, K. Yang, and L. Xie, “An End-to-End Point-Based Method and a New Dataset for Street-Level Point Cloud Change Detection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, 2023, doi: 10.1109/TGRS.2023.3295386.

[179] I. de Gélis, S. Lefèvre, and T. Corpetti, “Siamese KPConv: 3D multiple change detection from raw point clouds using deep learning,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 274–291, Mar. 2023, doi: 10.1016/j.isprsjprs.2023.02.001.

[180] L. Zhang, H. Wang, C. Liu, Z. Dong, and B. Yang, “ME-CPT: Multi-Task Enhanced Cross-Temporal Point Transformer for Urban 3D Change Detection,” Jan. 2025, [Online]. Available: http://arxiv.org/abs/2501.14004

[181] A. Kharroubi, F. Poux, Z. Ballouch, R. Hajji, and R. Billen, “Three Dimensional Change Detection Using Point Clouds: A Review,” *Geomatics*, vol. 2, no. 4, pp. 457–485, Oct. 2022, doi: 10.3390/geomatics2040025.

[182] A. Kharroubi, Z. Ballouch, I. Jeddoub, R. Hajji, and R. Billen, “Automated detection and structuration of building and vegetation changes from LiDAR point clouds,” *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-2/W8-2024, pp. 227–233, Dec. 2024, doi: 10.5194/isprs-archives-XLVIII-2-W8-2024-227-2024.

[183] A. Tamort, A. Kharroubi, R. Hajji, and R. Billen, “3D CHANGE DETECTION FOR SEMI-AUTOMATIC UPDATE OF BUILDINGS IN 3D CITY MODELS,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, International Society for Photogrammetry and Remote Sensing, Mar. 2024, pp. 349–355. doi: 10.5194/isprs-archives-XLVIII-4-W9-2024-349-2024.

[184] I. de Gélis, T. Corpetti, and S. Lefèvre, “Change detection needs change information: improving deep 3D point cloud change detection,” Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.12639

[185] L. Landrieuy and G. Obozinskiz, “Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs,” *SIAM J Imaging Sci*, vol. 10, no. 4, pp. 1724–1766, Nov. 2017, doi: 10.1137/17M1113436.

[186] Z. Xi and C. Hopkinson, “3D Graph-Based Individual-Tree Isolation (Treeiso) from Terrestrial Laser Scanning Point Clouds,” *Remote Sens (Basel)*, vol. 14, no. 23, Dec. 2022, doi: 10.3390/rs14236116.

[187] M. Weinmann, B. Jutzi, C. Mallet, and M. Weinmann, “GEOMETRIC FEATURES and THEIR RELEVANCE for 3D POINT CLOUD CLASSIFICATION,” in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus GmbH, May 2017, pp. 157–164. doi: 10.5194/isprs-annals-IV-1-W1-157-2017.

[188] P. Naylor, D. Di Carlo, A. Traviglia, M. Yamada, and M. Fiorucci, “Implicit neural representation for change detection,” Jul. 2023, [Online]. Available: http://arxiv.org/abs/2307.15428

[189] P. R. Hirt, Y. Xu, L. Hoegner, and U. Stilla, “Change Detection of Urban Trees in MLS Point Clouds Using Occupancy Grids,” *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 89, no. 4, pp. 301–318, Aug. 2021, doi: 10.1007/s41064-021-00179-4.

[190] Y. He *et al.*, “Deep Learning based 3D Segmentation: A Survey,” *Proceedings of*, vol. 1, no. 1, pp. 1–36, 2021, [Online]. Available: http://arxiv.org/abs/2103.05423

[191] R. Wahl and R. Klein, “Efficient RANSAC for Point-Cloud Shape Detection,” vol. 0, no. 0, pp. 1–12, 2007, doi: 10.1111/j.1467-8659.2007.01016.x.

[192] Q. H. Truong, “Knowledge-based 3D point clouds processing.” [Online]. Available: https://theses.hal.science/tel-00977434

[193] J. Ponciano, M. Roetner, A. Reiterer, and F. Boochs, “Object Semantic Segmentation in Point Clouds — Comparison of a Deep Learning and a Knowledge-Based Method,” 2021.

[194] M. F. Alkadri, S. Alam, H. Santosa, A. Yudono, and S. M. Beselly, “Investigating Surface Fractures and Materials Behavior of Cultural Heritage Buildings Based on the Attribute Information of Point Clouds Stored in the TLS Dataset,” *Remote Sens (Basel)*, vol. 14, no. 2, Jan. 2022, doi: 10.3390/rs14020410.

[195] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view Convolutional Neural Networks for 3D Shape Recognition.” [Online]. Available: http://vis-www.cs.umass.edu/mvcnn.

[196] A. Dai and M. Nießner, “3DMV: Joint 3D-Multi-View Prediction for 3D Semantic Scene Segmentation,” Mar. 2018, [Online]. Available: http://arxiv.org/abs/1803.10409

[197] A. Kundu *et al.*, “Virtual Multi-view Fusion for 3D Semantic Segmentation,” Jul. 2020, [Online]. Available: http://arxiv.org/abs/2007.13138

[198] A. Boulch, J. Guerry, B. Le Saux, and N. Audebert, “SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks,” *Comput Graph*, vol. 71, pp. 189–198, Apr. 2018, doi: 10.1016/J.CAG.2017.11.010.

[199] B. Wu, A. Wan, X. Yue, and K. Keutzer, “SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud,” Oct. 2017, [Online]. Available: http://arxiv.org/abs/1710.07368

[200] G. Karara, R. Hajji, and F. Poux, “3D point cloud semantic augmentation: Instance segmentation of 360◦ panoramas by deep learning techniques,” *Remote Sens (Basel)*, vol. 13, no. 18, Sep. 2021, doi: 10.3390/rs13183647.

[201] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, “RangeViT: Towards Vision Transformers for 3D Semantic Segmentation in Autonomous Driving,” Jan. 2023, [Online]. Available: http://arxiv.org/abs/2301.10222

[202] Y. Xu, X. Tong, and U. Stilla, “Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry,” Jun. 01, 2021, *Elsevier B.V.* doi: 10.1016/j.autcon.2021.103675.

[203] Z. Fang, B. Xiong, and F. Liu, “Sparse point-voxel aggregation network for efficient point cloud semantic segmentation,” *IET Computer Vision*, vol. 16, no. 7, pp. 644–654, Oct. 2022, doi: 10.1049/cvi2.12131.

[204] M. Ye, R. Wan, S. Xu, T. Cao, and Q. Chen, “DRINet++: Efficient Voxel-as-point Point Cloud Segmentation.”

[205] H. Li *et al.*, “MVPNet: A multi-scale voxel-point adaptive fusion network for point cloud semantic segmentation in urban scenes,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 122, Aug. 2023, doi: 10.1016/j.jag.2023.103391.

[206] H. Su *et al.*, “SPLATNet: Sparse Lattice Networks for Point Cloud Processing,” 2018, doi: 10.1109/CVPR.2018.00268.

[207] R. A. Rosu, P. Schütt, J. Quenzel, and S. Behnke, “LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices,” Dec. 2019, [Online]. Available: http://arxiv.org/abs/1912.05905

[208] R. A. Rosu, P. Schütt, J. Quenzel, and S. Behnke, “LatticeNet: Fast Spatio-Temporal Point Cloud Segmentation Using Permutohedral Lattices,” Aug. 2021, [Online]. Available: http://arxiv.org/abs/2108.03917

[209] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.02413

[210] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” Dec. 2016, [Online]. Available: http://arxiv.org/abs/1612.00593

[211] W. Wu, Z. Qi, and L. Fuxin, “PointConv: Deep Convolutional Networks on 3D Point Clouds,” Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.07246

[212] H. Thomas, Y.-H. H. Tsai, T. D. Barfoot, and J. Zhang, “KPConvX: Modernizing Kernel Point Convolution with Kernel Attention,” May 2024, [Online]. Available: http://arxiv.org/abs/2405.13194

[213] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “KPConv: Flexible and Deformable Convolution for Point Clouds.”

[214] Z. Zeng, Y. Xu, Z. Xie, J. Wan, W. Wu, and W. Dai, “RG-GCN: A Random Graph Based on Graph Convolution Network for Point Cloud Semantic Segmentation,” *Remote Sens (Basel)*, vol. 14, no. 16, Aug. 2022, doi: 10.3390/rs14164055.

[215] L. Landrieu and M. Simonovsky, “Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567, 2018, doi: 10.1109/CVPR.2018.00479.

[216] T. Jiang, J. Sun, S. Liu, X. Zhang, Q. Wu, and Y. Wang, “Hierarchical semantic segmentation of urban scene point clouds via group proposal and graph attention network,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 105, Dec. 2021, doi: 10.1016/j.jag.2021.102626.

[217] D. Lu, Q. Xie, M. Wei, K. Gao, L. Xu, and J. Li, “Transformers in 3D Point Clouds: A Survey,” May 2022, [Online]. Available: http://arxiv.org/abs/2205.07417

[218] X. Lai *et al.*, “Stratified Transformer for 3D Point Cloud Segmentation,” Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.14508

[219] J. Zhou, Y. Xiong, C. Chiu, F. Liu, and X. Gong, “SAT: Size-Aware Transformer for 3D Point Cloud Semantic Segmentation,” Jan. 2023, [Online]. Available: http://arxiv.org/abs/2301.06869

[220] B. Fei *et al.*, “Self-supervised Learning for Pre-Training 3D Point Clouds: A Survey,” May 2023, [Online]. Available: http://arxiv.org/abs/2305.04691

[221] H. and G. W. and W. C. and L. J. Lin Xi and Luo, “A Multi-task Learning Framework for Semantic Segmentation in MLS Point Clouds,” in *Artificial Intelligence and Security*, X. and X. Z. and B. E. Sun Xingming and Zhang, Ed., Cham: Springer International Publishing, 2022, pp. 382–392.

[222] A. Kirillov *et al.*, “Segment Anything,” Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.02643

[223] Y. Hong *et al.*, “3D-LLM: Injecting the 3D World into Large Language Models,” Jul. 2023, [Online]. Available: http://arxiv.org/abs/2307.12981

[224] U. Stilla and Y. Xu, “Change detection of urban objects using 3D point clouds: A review,” Mar. 01, 2023, *Elsevier B.V.* doi: 10.1016/j.isprsjprs.2023.01.010.

[225] G. R. Dini, K. Jacobsen, F. Rottensteiner, M. Al Rajhi, and C. Heipke, “3D BUILDING CHANGE DETECTION USING HIGH RESOLUTION STEREO IMAGES AND A GIS DATABASE,” 2012.

[226] M. Slavík, K. Kuželka, R. Modlinger, I. Tomášková, and P. Surový, “Uav laser scans allow detection of morphological changes in tree canopy,” *Remote Sens (Basel)*, vol. 12, no. 22, pp. 1–15, Nov. 2020, doi: 10.3390/rs12223829.

[227] U. Okyay, J. Telling, C. L. Glennie, and W. E. Dietrich, “Airborne lidar change detection: An overview of Earth sciences applications,” *Earth Sci Rev*, vol. 198, no. August, p. 102929, 2019, doi: 10.1016/j.earscirev.2019.102929.

[228] D. Lague, N. Brodu, and J. Leroux, “Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (N-Z),” *ISPRS*

Journal of Photogrammetry and Remote Sensing, vol. 82, pp. 10–26, 2013, doi: 10.1016/j.isprsjprs.2013.04.009.

[229] M. Awrangjeb, "Effective generation and update of a building map database through automatic building change detection from LiDAR point cloud data," *Remote Sens (Basel)*, vol. 7, no. 10, pp. 14119–14150, 2015, doi: 10.3390/rs71014119.

[230] C. Dai, Z. Zhang, and D. Lin, "An object-based bidirectional method for integrated building extraction and change detection between multimodal point clouds," *Remote Sens (Basel)*, vol. 12, no. 10, May 2020, doi: 10.3390/rs12101680.

[231] I. de Gélis, S. Lefèvre, and T. Corpetti, "Siamese KPConv: 3D multiple change detection from raw point clouds using deep learning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 197, pp. 274–291, Mar. 2023, doi: 10.1016/j.isprsjprs.2023.02.001.

[232] Z. Ballouch, I. Jeddoub, R. Hajji, J. P. Kasprzyk, and R. Billen, "Towards a Digital Twin of Liege: The Core 3D Model based on Semantic Segmentation and Automated Modeling of LiDAR Point Clouds," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Copernicus Publications, May 2024, pp. 13–20. doi: 10.5194/isprs-annals-X-4-W4-2024-13-2024.

[233] Z. Xi and C. Hopkinson, "3D Graph-Based Individual-Tree Isolation (Treeiso) from Terrestrial Laser Scanning Point Clouds," *Remote Sens (Basel)*, vol. 14, no. 23, Dec. 2022, doi: 10.3390/rs14236116.

[234] B. P. Rafamatanantsoa, I. Jeddoub, A. Yarroudh, R. Hajji, and R. Billen, "City2Twin: an open urban digital twin from data integration to visualization and analysis," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-2/W8-2024, pp. 387–394, Dec. 2024, doi: 10.5194/isprs-archives-XLVIII-2-W8-2024-387-2024.

[235] M. Soilán, A. Sánchez-Rodríguez, P. Del Río-Barral, C. Perez-Collazo, P. Arias, and B. Riveiro, "Review of laser scanning technologies and their applications for road and railway infrastructure monitoring," *Infrastructures (Basel)*, vol. 4, no. 4, 2019, doi: 10.3390/infrastructures4040058.

[236] D. Lamas, M. Soilán, J. Grandío, and B. Riveiro, "Automatic point cloud semantic segmentation of complex railway environments," *Remote Sens (Basel)*, vol. 13, no. 12, Jun. 2021, doi: 10.3390/rs13122332.

[237] X. Chen *et al.*, "Railway overhead contact system point cloud classification," *Sensors*, vol. 21, no. 15, Aug. 2021, doi: 10.3390/s21154961.

[238] X. Roynard, J. E. Deschaud, and F. Goulette, "Paris-lille-3D: A point cloud dataset for urban scene segmentation and classification," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2018-June, pp. 2108–2111, 2018, doi: 10.1109/CVPRW.2018.00272.

[239] W. Tan *et al.*, “Toronto-3D: A large-scale mobile LiDAR dataset for semantic segmentation of urban roadways2211,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2020-June, pp. 797–806, 2020, doi: 10.1109/CVPRW50498.2020.00109.

[240] J. Behley *et al.*, “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences,” Apr. 2019, [Online]. Available: http://arxiv.org/abs/1904.01416

[241] R. Zhang, Y. Wu, W. Jin, and X. Meng, “Deep-Learning-Based Point Cloud Semantic Segmentation: A Survey,” Sep. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/electronics12173642.

[242] K. Oh *et al.*, “A Review of Deep Learning Applications for Railway Safety,” Oct. 01, 2022, *MDPI*. doi: 10.3390/app122010572.

[243] S. Hoerbinger, M. Obriejetan, H. P. Rauch, and M. Immitzer, “Assessment of safety-relevant woody vegetation structures along railway corridors,” *Ecol Eng*, vol. 158, Dec. 2020, doi: 10.1016/j.ecoleng.2020.106048.

[244] M. Sturari, M. Paolanti, E. Frontoni, A. Mancini, and P. Zingaretti, “Robotic Platform for Deep Change Detection for Rail Safety and Security.”

[245] P. R. Hirt, Y. Xu, L. Hoegner, and U. Stilla, “Change Detection of Urban Trees in MLS Point Clouds Using Occupancy Grids,” *PFG - Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 89, no. 4, pp. 301–318, Aug. 2021, doi: 10.1007/s41064-021-00179-4.

[246] M. Arastounia, “Automated recognition of railroad infrastructure in rural areas from LIDAR data,” *Remote Sens (Basel)*, vol. 7, no. 11, pp. 14916–14938, 2015, doi: 10.3390/rs71114916.

[247] G. Vosselman and R. Klein, “Visualisation and structuring of point clouds,” in *Airborne and terrestrial laser scanning*, M. G. Vosselman and H. G. Maas, Eds., CRC Press (Taylor & Francis), 2010, pp. 45–81.

[248] M. Muja and D. G. Lowe, “Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration,” in *VISAPP 2009 - Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, February 5-8, 2009 - Volume 1*, A. Ranchordas and H. Araújo, Eds., INSTICC Press, 2009, pp. 331–340.

[249] L. Chen, J. Jung, and G. Sohn, “Multi-Scale Hierarchical CRF for Railway Electrification Asset Classification from Mobile Laser Scanning Data,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 12, no. 8, pp. 3131–3148, Aug. 2019, doi: 10.1109/JSTARS.2019.2918272.

[250] J. Grandio, B. Riveiro, M. Soilán, and P. Arias, “Point cloud semantic segmentation of complex railway environments using deep learning,” *Autom Constr*, vol. 141, Sep. 2022, doi: 10.1016/j.autcon.2022.104425.

[251] B. Ton, F. Ahmed, and J. Linssen, “Semantic Segmentation of Terrestrial Laser Scans of Railway Catenary Arches: A Use Case Perspective,” *Sensors*, vol. 23, no. 1, Jan. 2023, doi: 10.3390/s23010222.

[252] O. Zendel, M. Murschitz, M. Zeilinger, D. Steininger, S. Abbasi, and C. Beleznai, “RailSem19: A Dataset for Semantic Rail Scene Understanding.” [Online]. Available: www.wilddash.cc

[253] J. Harb, N. Rébéna, R. Chosidow, G. Roblin, R. Potarusov, and H. Hajri, “FRSign: A Large-Scale Traffic Light Dataset for Autonomous Trains,” Feb. 2020, [Online]. Available: http://arxiv.org/abs/2002.05665

[254] T. Toprak, B. Aydın, B. Belenlioğlu, C. Güzeliş, and M. A. Selver, “Railway Pedestrian Dataset (RAWPED),” Apr. 2020, *Zenodo*. doi: 10.1109/TVT.2020.2983825.

[255] T. Toprak, B. Belenlioglu, B. Aydin, C. Guzelis, and M. A. Selver, “Conditional Weighted Ensemble of Transferred Models for Camera Based Onboard Pedestrian Detection in Railway Driver Support Systems,” *IEEE Trans Veh Technol*, vol. 69, no. 5, pp. 5041–5054, May 2020, doi: 10.1109/TVT.2020.2983825.

[256] P. Leibner, F. Hampel, and C. Schindler, “GERALD: A novel dataset for the detection of German mainline railway signals,” *Proc Inst Mech Eng F J Rail Rapid Transit*, vol. 0, no. 0, p. 09544097231166472, doi: 10.1177/09544097231166472.

[257] R. Tagiew *et al.*, “OSDaR23: Open Sensor Data for Rail 2023,” May 2023, doi: 10.1109/ICRAE59816.2023.10458449.

[258] B. Qiu *et al.*, “WHU-Railway3D: A Diverse Dataset and Benchmark for Railway Point Cloud Semantic Segmentation,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 12, pp. 20900–20916, Dec. 2024, doi: 10.1109/TITS.2024.3469546.

[259] B. Ton, “Labelled high resolution point cloud dataset of 15 catenary arches in the Netherlands,” 2022, *4TU.ResearchData*. doi: 10.4121/17048816.v1.

[260] R. Tilly *et al.*, “Open Sensor Data for Rail 2023,” 2023, *TIB*. doi: 10.57806/9MV146R0.

[261] M. Eastepp, L. Faris, and K. Ricks, “UA_L-DoTT: University of Alabama’s large dataset of trains and trucks,” *Data Brief*, vol. 42, Jun. 2022, doi: 10.1016/j.dib.2022.108073.

[262] G. D’Amico *et al.*, “TrainSim: A Railway Simulation Framework for LiDAR and Camera Dataset Generation,” Feb. 2023, [Online]. Available: http://arxiv.org/abs/2302.14486

[263] A. R. Fayjie and P. Vandewalle, “Few-Shot Learning on Point Clouds for Railroad Segmentation,” in *IS and T International Symposium on Electronic Imaging Science and Technology*, Society for Imaging Science and Technology, 2023. doi: 10.2352/EI.2023.35.17.3DIA-100.

[264] Y. Wang, “Railway SLAM Dataset,” 2022, *IEEE Dataport*. doi: 10.21227/kq9s-ct48.

[265] M. Corongiu *et al.*, “Classification of railway assets in mobile mapping point clouds,” vol. XLIII, pp. 219–225, 2020.

[266] J. L. A. Riquelme *et al.*, “3D Point Cloud of a railway slope - MOMIT (Multi- scale observation and monitoring of railway infrastructure threats) EU project - H2020-EU.3.4.8.3. - Grant agreement ID: 777630,” Apr. 2020, *Zenodo*. doi: 10.5281/zenodo.3777996.

[267] M. Cserep, “Hungarian MLS point clouds of railroad environment and annotated ground truth data,” 2022, *Mendeley Data*. doi: 10.17632/ccxpzhx9dj.1.

[268] T. Jiang *et al.*, “RailPC: A large-scale railway point cloud semantic segmentation dataset,” *CAAI Trans Intell Technol*, Dec. 2024, doi: 10.1049/cit2.12349.

[269] M. Abid, M. Teixeira, A. Mahtani, and T. Laurent, “RailCloud-HdF: A Large-Scale Point Cloud Dataset for Railway Scene Semantic Segmentation,” in *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Science and Technology Publications, Lda, 2024, pp. 159–170. doi: 10.5220/0012394800003660.

[270] S. M. González-Collazo, J. Balado, E. González, and A. Nurunnabi, “A discordance analysis in manual labelling of urban mobile laser scanning data used for deep learning based semantic segmentation,” *Expert Syst Appl*, vol. 230, Nov. 2023, doi: 10.1016/j.eswa.2023.120672.

[271] G. Girardeau-Montaut, “CloudCompare,” 2010. [Online]. Available: https://www.cloudcompare.org/

[272] E. Sevgen and S. Abdikan, “Classification of Large-Scale Mobile Laser Scanning Data in Urban Area with LightGBM,” *Remote Sens (Basel)*, vol. 15, no. 15, Aug. 2023, doi: 10.3390/rs15153787.

[273] M. Letard *et al.*, “3DMASC: Accessible, explainable 3D point clouds classification. Application to Bi-Spectral Topo-Bathymetric lidar data.” [Online]. Available: https://hal.science/hal-04072068

[274] S. Lytkin *et al.*, “Saint Petersburg 3D: Creating a Large-Scale Hybrid Mobile LiDAR Point Cloud Dataset for Geospatial Applications,” *Remote Sens (Basel)*, vol. 15, no. 11, Jun. 2023, doi: 10.3390/rs15112735.

[275] Y. Li, C. Fan, X. Wang, and Y. Duan, “SPNet: Multi-Shell Kernel Convolution for Point Cloud Semantic Segmentation,” Sep. 2021, [Online]. Available: http://arxiv.org/abs/2109.11610

[276] H. Thomas, J.-E. Deschaud, B. Marcotegui, F. Goulette, and Y. Le Gall, “Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods,” Aug. 2018, doi: 10.1109/3DV.2018.00052.

[277] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview,” Aug. 2020, [Online]. Available: http://arxiv.org/abs/2008.05756

[278] E. González, J. Balado, P. Arias, and H. Lorenzo, “Realistic correction of sky-coloured points in Mobile Laser Scanning point clouds,” *Opt Laser Technol*, vol. 149, May 2022, doi: 10.1016/j.optlastec.2021.107807.

[279] Z. Xia *et al.*, “SCPNet: Semantic Scene Completion on Point Cloud.” [Online]. Available: https://codalab.lisn.upsaclay.fr/competitions/7170#results

[280] Y. Feng, Y. Xu, Y. Xia, C. Brenner, and M. Sester, “Gap completion in point cloud scene occluded by vehicles using SGC-Net,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 215, pp. 331–350, Sep. 2024, doi: 10.1016/j.isprsjprs.2024.07.009.

[281] T. Czerniawski, J. W. Ma, and Fernanda Leite, “Automated building change detection with amodal completion of point clouds,” *Autom Constr*, vol. 124, Apr. 2021, doi: 10.1016/j.autcon.2021.103568.

[282] T. H. Nguyen Son H. and Kolbe, “Identification and Interpretation of Change Patterns in Semantic 3D City Models,” in *Recent Advances in 3D Geoinformation Science*, A. and B. C. Kolbe Thomas H. and Donaubauer, Ed., Cham: Springer Nature Switzerland, 2024, pp. 479–496.

CURRICULUM

Abderrazzaq Kharroubi was born in Morocco (1996). He earned a State Engineer degree in Topography from the Institut Agronomique et Vétérinaire Hassan II, Rabat, in 2019. His interest in geomatics and 3D data processing led him to pursue research in this field.

In December 2019, he joined the Geomatics and Geometrology Unit (now GeoScITY) at the University of Liège as a research assistant. He worked on the Terra Mosana project. Then, he participated in 3D surveys of the Coudenberg Museum in Brussels using laser scanning and photogrammetry. Later, he contributed to the Promone project, where he documented the Carolingian Oratory of Germigny-des-Prés and developed machine learning algorithms to classify historical mosaics automatically.

In October 2021, he started his PhD under the supervision of Prof. Roland Billen, funded by the Fonds de la Recherche Scientifique (F.R.S.-FNRS). His research focuses on processing and analyzing 3D point clouds, with applications in change detection and city modeling. His work involves developing methods to detect and classify changes in LiDAR data, particularly for monitoring urban areas and railway networks.

Alongside his PhD, he has taken on senior scientific roles in applied projects. He was the scientific advisor on TrackGen, a project that aimed to automate the creation of realistic 3D railway simulation environments using AI and geodata. He is currently a senior scientific lead on WaLoD2, a project focused on 3D reconstruction of buildings from LiDAR data at Level of Detail 2.2 over the Wallonia, Belgium.

He has also been involved in teaching and supervising MSc and BSc students at the University of Liège. His work balances research and practical applications, contributing to both academic studies and real-world projects in 3D mapping and 3D geodata processing.

LIST OF PUBLICATIONS

International Journal Papers (ISI)

Kharroubi A, Remondino F, Ballouch Z, Hajji R, Billen R. Semantic and Geometric Fusion for Object-Based 3D Change Detection in LiDAR Point Clouds. Remote Sensing. 2025, 17(7), 1311; https://doi.org/10.3390/rs17071311

Kharroubi A, Ballouch Z, Hajji R, Yarroudh A, Billen R. Multi-Context Point Cloud Dataset and Machine Learning for Railway Semantic Segmentation. Infrastructures. 2024; 9(4):71. https://doi.org/10.3390/infrastructures9040071

Ballouch Z, Hajji R, **Kharroubi A**, Poux F, Billen R. Investigating Prior-Level Fusion Approaches for Enriched Semantic Segmentation of Urban LiDAR Point Clouds. Remote Sensing. 2024; 16(2):329. https://doi.org/10.3390/rs16020329

Ballouch Z, Hajji R, Poux F, **Kharroubi A**, Billen R. A Prior Level Fusion Approach for the Semantic Segmentation of 3D Point Clouds Using Deep Learning. Remote Sensing. 2022; 14(14):3415. https://doi.org/10.3390/rs14143415

Kharroubi A, Poux F, Ballouch Z, Hajji R, Billen R. Three Dimensional Change Detection Using Point Clouds: A Review. Geomatics. 2022; 2(4):457-485. https://doi.org/10.3390/geomatics2040025

El Barhoumi N, Hajji R, Bouali Z, Ben Brahim Y, **Kharroubi A**. Assessment of 3D Models Placement Methods in Augmented Reality. Applied Sciences. 2022; 12(20):10620. https://doi.org/10.3390/app122010620

International Refereed Conferences

Kharroubi, A., Ballouch, Z., Jeddoub, I., Hajji, R., and Billen, R.: Automated detection and structuration of building and vegetation changes from LiDAR point clouds, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-2/W8-2024, 227–233, https://doi.org/10.5194/isprs-archives-XLVIII-2-W8-2024-227-2024, 2024.

Yarroudh, A., **Kharroubi, A.**, Jeddoub, I., Ballouch, Z., and Billen, R.: Railway reconstruction from 3D point cloud using Deep Learning and Parametric Modeling, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-2/W8-2024, 477–482, https://doi.org/10.5194/isprs-archives-XLVIII-2-W8-2024-477-2024, 2024.

Yarroudh, A., **Kharroubi, A.**, Jeddoub, I., and Billen, R.: Automatic upgrade of 3D building models to LoD3 using 3D Point Clouds and Grounding DINO, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-2/W8-2024, 471–476, https://doi.org/10.5194/isprs-archives-XLVIII-2-W8-2024-471-2024, 2024.

Tamort, A., **Kharroubi, A.**, Hajji, R., and Billen, R.: 3D change detection for semi-automatic update of buildings in 3d city models, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-4/W9-2024, 349–355, https://doi.org/10.5194/isprs-archives-XLVIII-4-W9-2024-349-2024, 2024.

Yarroudh, A., **Kharroubi, A.**, Billen, R.: Optim3D: Efficient and Scalable Generation of Large-Scale 3D Building Models. In: Kolbe, T.H., Donaubauer, A., Beil, C. (eds) Recent Advances in 3D Geoinformation Science. 3DGeoInfo 2023. Lecture Notes in Geoinformation and Cartography. Springer, Cham. https://doi.org/10.1007/978-3-031-43699-4_50 , 2024

Hajji, R., **Kharroubi, A.**, Ben Brahim, Y., Bahhane, Z., and El Ghazouani, A.: Integration of BIM and mobile augmented reality in the AECO domain, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVI-4/W3-2021, 131–138, https://doi.org/10.5194/isprs-archives-XLVI-4-W3-2021-131-2022, 2022

Kharroubi, A., Van Wersch, L., Billen, R., and Poux, F.: Tesserae3D: a benchmark for tesserae semantic segmentation in 3D point clouds, ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., V-2-2021, 121–128, https://doi.org/10.5194/isprs-annals-V-2-2021-121-2021, 2021

Nys, G.-A., **Kharroubi, A.**, Poux, F., and Billen, R.: An extension of CityJSON to support point clouds, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLIII-B4-2021, 301–306, https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-301-2021, 2021

Kharroubi, A., Billen, R., and Poux, F.: Marker-less mobile augmented reality application for massive 3d point clouds and semantics, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLIII-B2-2020, 255–261, https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-255-2020, 2020

Kharroubi, A., Hajji, R., Billen, R., and Poux, F.: Classification and integration of massive 3D points clouds in a virtual reality (VR) environment, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-2/W17, 165–171, https://doi.org/10.5194/isprs-archives-XLII-2-W17-165-2019, 2019