



Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C

Dissertationen

Heft Nr. 915

Eike Ruben Barnefske

**Automated Segmentation and Classification with
Artificial Neural Networks of Objects in 3D Point Clouds**

München 2023

Bayerische Akademie der Wissenschaften

ISSN 0065-5325

ISBN 978-3-7696-5327-4

Diese Arbeit ist gleichzeitig veröffentlicht in:
repOS - Das Open Science Repository der HCU Hamburg
<https://nbn-resolving.org/urn:nbn:de:agb:3A1373-repos-11804>, Hamburg 2023

Automated Segmentation and Classification with Artificial Neural Networks
of Objects in 3D Point Clouds

Approved Dissertation

to obtain the academic degree Doktor-Ingenieur (Dr.-Ing.)
submitted to the HafenCity Universität Hamburg
in the field of Geodesy and Geoinformatics

by

Eike Ruben Barnefske

München 2023

Bayerische Akademie der Wissenschaften

Adresse der DGK:



Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)

Alfons-Goppel-Straße 11 • D – 80 539 München
Telefon +49 - 331 - 288 1685 • E-Mail post@dgk.badw.de
<http://www.dgk.badw.de>

Prüfungskommission:

Chairperson: Prof. Dr.-Ing. Ingo Weidlich

Supervisors: Prof. Dr.-Ing. Harald Sternberg

Prof. Dr.-Ing. Alexander Reiterer (Albert-Ludwigs-Universität Freiburg)

Prof. Dr.-Ing. Jochen Schiewe

Day of Defence: July 20, 2023

© 2023 Bayerische Akademie der Wissenschaften, München

Alle Rechte vorbehalten. Ohne Genehmigung der Herausgeber ist es auch nicht gestattet,
die Veröffentlichung oder Teile daraus auf photomechanischem Wege (Photokopie, Mikrokopie) zu vervielfältigen

ISSN 0065-5325

ISBN 978-3-7696-5327-4

Danksagung

Zu allererst möchte mich bei meinem Doktorvater Harald Sternberg für den stets offenen und vertrauensvollen Austausch und die Unterstützung, insbesondere, wenn es schwierig wurde, bedanken. Danke für dein Vertrauen in mich.

Für die vielen guten Anmerkungen zur Forschung, die Feedbacks, die Ermutigungen, die Impulse und die Diskussionen möchte ich mich weiterhin bei Alexander Reiterer bedanken.

Ein großer Dank geht an Annette Scheider, Meike Ahrend und Clemens Semmelroth, die mich nicht nur bei fachlichen Fragen, bei den durchgeführten Untersuchungen und den Korrekturen, sondern auch durch viele bestärkende und ermutigende Gespräche unterstützt haben.

Bedanken möchte ich mich bei meinen Kolleginnen und Kollegen der Arbeitsgruppe und des geodätischen Labors für das gute Arbeitsumfeld, die vielen kleinen Hilfestellungen, euch Offenheit und das schöne Gemeinschaftsgefühl.

Ein besonderer Dank geht an meine Eltern Dieter und Gabrielle, meine Großeltern Oskar und Hella, meine Schwester Lena und meine Freundin Sarah für die unzähligen Momente, in denen ihr mich unterstützt habt, die vielen aufbauenden Gespräche und das nie nachlassende Vertrauen in mich.

Bei Almut, Katharina, Lars und Jochen möchte ich mich dafür bedanken, dass ihr mich auch mal vom Schreibtisch weglocken konntet. Mit und bei euch konnte ich bei Kuchennachmittagen, auf unseren Reisen und bei den vielen tollen Events die Kraft tanken, die ich brauchte.

Zuletzt möchte ich mich noch bei den Kameraden der Freiwilligen Feuerwehr Fuhlsbüttel dafür bedanken, dass ich von euch den nötigen Rückhalt, das Verständnis und die Bestärkung für das Beenden der Arbeit erfahren durfte.

Abstract

The recording of objects surfaces with *Light Imaging, Detection and Ranging* (LIDAR) scanners is a well-established surveying method for the highly accurate and detailed geometric creation of models. The result of LIDAR recordings is a three-dimensional (3D) point cloud with geometric and spectral (intensity and color values) features that represent a geometric model of reality. This model is usually automatically extended by the human imagination with semantic information by looking at it, so that object classes, individual objects or measurement errors in the point cloud can be reliably identified. The easy interpretation of point cloud scenes and its effective recording with LIDAR scanners has led to the fact that point clouds become a quasi-format standard for 3D models, besides to mesh, voxel and parametric models. Semantic features are necessary for automatic processing of point clouds, for example, in a building information model. Currently, semantic enhancement of point cloud information is mostly done manually, and automation (e.g., via deep learning methods) is still a subject of research. In particular, *Artificial Neural Networks* (ANN) have proven to be effective for this task when the data and hyperparameters (HPs) are optimized.

In this thesis, the *PointNet* ANN was used as an example to research which are optimal point cloud data and HPs. The creation of training data with manual annotation tools, the implementation and research of processes for automatic semantic segmentation, and the development of a heuristic quality model for the evaluation of point cloud datasets and of semantic segmentation processes are the central research issues. The annotation tool, *Point Cloud Classification Tools* (PCCT), was developed to investigate automation, training processes of annotators, and features influence. For automatic point cloud processing, influences are points from erroneous measurements, the class inequality and the semantic class definitions. Different class definitions and methods for minimizing the differences in class sizes have been developed, adaptations in point cloud pre-processing have been applied and the weighting of infrequent classes have been optimized.

The research results show that optimal (data-based) HPs for semantic segmentation of a building dataset can be defined. This HP set and the approach can be used as guidelines for similar projects. An increase in recall of more than 50% for infrequently occurring classes can be achieved by algorithm-based class definition and class size consideration. Using the heuristic quality model, available training data and semantic segmentations can be evaluated and compared.

Zusammenfassung

Die flächenhafte Erfassung von Objektoberflächen mit *Light imaging, detection and ranging* (LIDAR) Scannern ist ein etabliertes Vermessungsverfahren zur hoch-genauen und detailreichen geometrischen Erstellung von Modellen. Das Ergebnis der LIDAR Erfassung ist eine dreidimensionale (3D) Punktwolke mit geometrischen und spektralen Merkmalen, die ein geometrisches Modell der Realität darstellen. Dieses Modell kann durch Menschen beim Betrachten meist automatisch um semantische Informationen erweitert werden, so dass Objektklassen, einzelne Objekte oder Messfehler in der Punktwolke sicher erkannt werden. Die einfache Interpretation durch den Menschen von Punktwolkenszenen und deren effektiven Erfassung mit LIDAR Scannern hat dazu geführt, dass Punktwolken neben den Mesh-, den Voxel- und den parametrischen Modellen quasi zu einem Formatstandard geworden sind. Semantische Merkmale sind für die automatische Verarbeitung der Punktwolken, z. B. in einem Bauwerksinformationsmodell, notwendig. Die semantische Erweiterung der Punktwolkeninformationen wird aktuell meist händisch durchgeführt und eine Automatisierung (z. B. mittels Deep Learning Verfahren) ist Gegenstand der Forschung. Insbesondere haben sich für diese Aufgabe *Künstliche Neuronale Netze* (KNN) als effektiv erwiesen, wenn die Daten und Hyperparameter optimiert sind.

In dieser Arbeit wurde am Beispiel des KNN *PointNet* erforscht, welche Punktwolkendaten und Hyperparameter optimal sind. Die Erstellung von Trainingsdaten mit händischen Annotationswerkzeugen, die Implementierung und Erforschung von Prozessen zur automatischen semantischen Segmentierung, sowie die Entwicklung eines heuristischen Qualitätsmodells zur Evaluation von Punktwolkendatensätzen und von semantischen Segmentierungsprozessen standen im Fokus. Das Annotationswerkzeug *Point Cloud Classification Tools* (PCCT) wurde entwickelt, mit dem die Automatisierung, die Trainingsprozesse von Annotatoren und die Funktionen in Annotationswerkzeugen untersucht werden. Bei der automatischen Punktwolkenverarbeitung sind die Einflüsse Punkte aus fehlerhaften Messungen, Klassenungleichheit und die semantische Klassendefinition zu berücksichtigen. Verschiedene Klassendefinitionen und Methoden für die Minimierung der unterschiedlichen Klassengrößen wurden entwickelt, Adaptionen bei der Punktwolkenvorverarbeitung wurden angewendet und die Gewichtung von seltenen Klassen wurde optimiert.

Die Forschungsergebnisse zeigen, dass optimale (datenbasierte) Hyperparameter für die semantische Segmentierung eines Bauwerksdatensatzes definiert werden können. Diese Hyperparameter und das Vorgehen können als Richtlinien für ähnliche Projekte verwendet werden. Eine Steigerung der semantischen Genauigkeit um bis 50% (Recall) ist bei selten vorkommenden Klassen kann durch eine algorithmusbezogene Klassendefinition und die Berücksichtigung der Klassengrößen erzielt werden. Mittels des heuristischen Qualitätsmodells können verfügbare Trainingsdaten und semantische Segmentierungen evaluiert werden.

Table of Contents

Danksagung	i
Abstract	iii
Zusammenfassung	v
List of Abbreviations	ix
List of Figures	xii
List of Tables	xvi
1 Introduction	1
1.1 Motivation	1
1.2 Research gaps in semantic segmentation of point clouds	3
1.3 Research objectives and questions	4
1.4 Outline of the thesis	6
2 State of the art	7
2.1 Recording systems for point clouds	7
2.2 Big data and machine learning	10
2.2.1 Data	11
2.2.2 Data pre-processing	12
2.2.3 Clustering and machine learning	13
2.2.4 Deep learning	15
2.2.5 Evaluation scheme and metrics	18
2.3 Manual semantic segmentation for point clouds	20
2.4 Training data for point cloud applications	24
2.5 Machine learning methods for point clouds	25
2.6 Deep learning methods for point clouds	28
2.6.1 Semantic segmentation with 2D projection-based deep learning methods	29
2.6.2 Semantic segmentation with 3D grid-based deep learning methods . .	32
2.6.3 Semantic segmentation with 3D point-based deep learning methods .	34
3 Connections of research publications	39
3.1 PAPER 0: PCCT: A point cloud classification tool to create 3D training data to adjust and develop 3D ConvNet	39
3.2 PAPER 1: Classification of erroneously measured points in 3D point clouds with ConvNet	40
3.3 PAPER 2: Evaluating the quality of semantic segmented 3D point clouds . .	40

3.4	PAPER 3: Evaluation of class distribution and class combinations on semantic segmentation of 3D point clouds with <i>PointNet</i>	42
3.5	Connections between the publications	42
4	Evaluation of the research results	45
4.1	Development and evaluation of a tool for manual point cloud annotation . . .	45
4.1.1	Survey of annotation tools for point clouds	45
4.1.2	Annotation tools for indoor terrestrial laser scanning point clouds . . .	47
4.1.3	Development of an annotation tool	49
4.1.4	Human factor in semantic segmentations of point clouds	51
4.2	Development of a quality model for heuristically describing semantic point clouds	52
4.2.1	Point cloud quality	53
4.2.2	Quality model for point clouds	55
4.2.3	Application and evaluation of the quality model for building point clouds	56
4.3	Development of a workflow for semantic segmentation to investigate datasets and point features as influences	58
4.3.1	Workflow development and algorithm selection	58
4.3.2	Hyperparameter selection for point-based deep learning methods . .	60
4.3.3	Data pre-processing and data influence	62
5	Conclusion and outlook	66
5.1	Conclusion	66
5.2	Outlook	68
5.2.1	Input format	68
5.2.2	Hand-crafted feature selection	69
5.2.3	Point cloud quality assessment	70
	Bibliography	I
	Appendices	XXVII
A	Peer-reviewed publications	XXVII
A.1	Klassifizierung von fehlerhaft gemessenen Punkten in 3D-Punktwolken mit ConvNet	XXVII
A.2	Evaluating the Quality of Semantic Segmented 3D Point Clouds	XLI
A.3	Evaluation of Class Distribution and Class Combinations on Semantic Segmentation of 3D Point Clouds with PointNet	LXXXIII
B	Non-peer-reviewed publication	CV
B.1	PCCT: A Point Cloud Classification Tool To Create 3D Training Data To Adjust And Develop 3D ConvNet	CV

List of Abbreviations

2D	Two-dimensional
3D	Three-dimensional
ALS	Air-born Laser Scanning
ANN	Artificial Neural Networks
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BB	Bounding Box
BEV	Bird's Eye View
CAD	Computer Aided Design
CNN	Convolutional Neural Networks
Conv layer	Convolutional layer
CRF	Conditional Random Field
CWS	Crowd-Working-Service
DHP	Data hyperparameter
DIM	Dense Image Matching
DL	Deep Learning
DT	Decision Tree
EDN	Encoder-Decoder-Network
EL	Ensemble Learning
EN	Encoder-Network
F-map	Feature-map
FC	Fully Connected
FCSOR	Fast Cluster Statistical Outlier Removal
FF	Feed-forward
FN	False Negative
FP	False Positive
FPS	Farthest Point Sampling
GIS	Geographic Information System
GNN	Graph Neural Network
GT	Ground Truth
HCU	HafenCity University
HP	Hyperparameter

List of Abbreviations

HT	Hough Transformation
IFC	Industry Foundation Class
IoU	Intersection over Union
kNN	k-Nearest-Neighbors
LIDAR	Light Imaging, Detection and Ranging
LoA	Level of Accuracy
LR	Learning Rate
LSA	Local Spatial Aware
LTSM	Long Term Short Memory
ML	Machine Learning
MLP	Multi Layer Perceptron
MSS	Multi Sensor System
OA	Overall Accuracy
OT	Offline Tool
PCA	Principal Component Analysis
PCCT	Point Cloud Classification Tool
RANSAC	.	Random Sample Consensus
RF	Random Forest
RG	Regional Growing
RGB	Red, Green, Blue
RGB-D	...	Red, Green, Blue and Depth
RGP	Research Gap
RNN	Recurrent Neural Networks
RO	Research Objective
ROS	Robot Operating System
RQ	Research Question
SC	Sparse Connected
SfM	Structure from Motion
SIFT	Scale-Invariant-Feature-Transformation
SL	Structured Light
SO	Self-organizing
SOR	Statistical Outlier Removal
SQL	Structured Query Language
SVM	Support-Vector-Machines
TL	Transfer Learning

List of Abbreviations

TLS Terrestrial Laser Scanning

TN True Negative

ToF Time of Flight

TP True Positive

VR Virtual Reality

WS Web Service

WV Windshield View

List of Figures

1	Measured TLS point cloud with segmentation and annotation errors. Class <i>Erroneous points</i> in red and class <i>Object</i> in blue.	4
2	Process of creating semantic point clouds. Recording of point clouds with an optical recording system. Registration of the individual recordings, resulting in a complete point cloud. Semantic segmentation according to given classes set. Modeling of objects in the semantic point cloud. Implementation of the models into an application. Taken from [62] and adapted.	7
3	Types of recording systems and the categorization by similar functional principles. Active sensors (radar, LIDAR, RGB-D) send out a signal and determine the distance to the object on reception. Passive sensors such as cameras use multiple images and corresponding points. Systemic data are surface models converted into the point cloud format. Taken from PAPER 2.	8
4	Point cloud of a TLS measurement with measurement errors, noise and non-uniform point density. Taken from [62] and adapted.	9
5	Four stages of the process for analyzing data. Inspired by [85, 86]	10
6	Definition of variables, objects and features.	12
7	Different classification types: a) raw point cloud, b) clustering, c) semantic segmentation, and d) instance segmentation.	13
8	Summary clustering and semantic similarity segmentation methods. Knowledge-based clustering methods (black), data-based clustering methods (green), and data-based semantic segmentation methods (orange).	14
9	Neuron and network architecture: a) Neuron architecture and function. a) Simple ANN with input and output layers. b) ANN with a hidden layer. Inspired by [119, 120]	16
10	Function of the one Conv layer at a CNN. Inspired by [120].	17
11	Common CNN-Architectures for semantic segmentation. a) Encoder-Network and b) Encoder-Decoder-Network	17
12	ANN with a recurrent layer. The outputs of the recurrent layer is used as additional input in the next pass. With time the inputs become less meaningful, so that its influence is lowered via weights.	18
13	Confusion matrix for the example of three classes. TP = <i>true positive</i> , FP = <i>false positive</i> and FN = <i>false negative</i> . TP of the one classes is equal to <i>true negative</i> (TN) for all other classes.	19
14	Requirements for a point classification tool.	20
15	Characteristics of semantic point cloud datasets with main attributes.	24

16	Methods for clustering and semantic segmentation. Left: Methods from ML for direct semantic segmentation. Right: Methods for clustering based on feature similarities.	25
17	Different graphs used as intermediate step for semantic segmentations. a) kNN-graph with $k = 4$ nearest geometric neighbors. b) Minimum spanning tree by feature value. c) Graph with all adjacent connections for each point. Color of the point indicates the feature value.	26
18	Workflow for semantic segmentation with ML. For non-DL methods, the initial features are important for the success of the semantic segmentation method (green boxes). In unsupervised methods, clustering and classification are separated commonly in two steps (yellow boxes). In supervised learning, it is usually done in one step (orange box). Inspired by [94].	27
19	Method characterization of DL network architectures for semantic segmentations by input formats and applied architectures.	28
20	BEV projection. The point cloud is oriented along the z-axis and transformed into a raster plan with fix a fix raster structure.	30
21	Projection of a 3D point cloud into (2D) image. a) Spherical or cylindrical projection. b) Multi-view-image projection and transformation.	31
22	Projection of the points onto a tangent plane. Creation of a multi-view image (Simplified 2D illustration).	31
23	Workflow semantic segmentation utilizing 3D grid structures. Top: The entire point cloud is transferred to one grid. Iteratively, several voxels are fed into a 3D CNN. The voxel grid is semantically segmented. The information is passed by interpolations to the point cloud. Bottom: A sub-voxel grid is created for each point. Each sub-voxel grid is classified by the 3D CNN. Each point is directly assigned to one class.	33
24	Voxel structures (in 2D perspective). a) Regular voxel grid. b) Occupancy grid. c) Octree with refinement due to point cloud density.	33
25	Structure and functions <i>PointNet++</i> . Encoding of point features in an iterative process considering the local neighborhood: (1) Selecting n points that are maximal wide away from each other. (2) Grouping of the features in the neighborhoods. (3) Applying a <i>PointNet</i> layer to feature extraction. (4) Repeating this process. Decoding by joint and step-wise interpolation of the features. (5) Classification layer at the end. Figure from [233].	35
26	Two methods for creating neighborhood input blocks. a) Fixed block size with the blocks sharing features via RNN. b) Variable block size with different fixed or dynamic radii. Used in CNN or MLP architectures. Inspired by [233, 234].	35

27	Overview of the connections of the four research publications. PAPER 0: Development of a browser-based classification tool. PAPER 1: Development of a workflow for semantic segmentation with <i>PointNet</i> and investigations on the influence of class <i>Erroneous points</i> . PAPER 2: Development of a quality model for the creation and evaluation of semantic point clouds. This quality model is used for the evaluation of the tool in PAPER 0 and the workflow in PAPER 3. PAPER 3: Extension of the workflow from PAPER 1 and development of methods to optimize the dataset for DL applications. Investigations of the development on the <i>PointNet</i> algorithm.	43
28	Central issues for improvement in available point cloud annotation tools: Data security, multi-user-capability, segmentation and classification functions, and automation of sub-operation steps.	49
29	Process of semantic segmentation of point clouds serving as an abstract model of the reality. Taken from PAPER 2 and adapted.	54
30	Quality model for semantic enhanced point clouds. Seven relevant characteristics with descriptive quality parameters are shown. Classification of necessary parameters for: Manual segmentations (filled blue circles), manual training data generation (unfilled blue circles) and automatic semantic segmentation (filled green circles).	55
31	Converting the quality model into an evaluation matrix for use on datasets, annotation tools, automatic semantic segmentation, and development monitoring.	57
32	Concept for a workflow to apply DL methods for semantic point cloud segmentation. Three modes for the procedure: Training, evaluation and application.	59
33	DHPs for semantic point clouds. The DHPs can be distinguished according to structural, semantic, geometric and spectral characteristics. A selection of the most common DHPs for each property is summarized.	62
34	Semantic segmentation accuracy (IoU) of four common network architectures for the dataset: <i>Semantic3d.net</i> [40]. Selection of four from eight classes of this dataset. The class <i>Scanning Artifacts</i> , which is equal to the class <i>Erroneous point</i> , can be detected poorly compared to the larger classes. Values are taken from the leader board of [40].	63
35	Comparison of semantic accuracy (<i>recall</i> and <i>precision</i>) on the point cloud of the HafenCity (outdoor) dataset: a) Without the class <i>Erroneous points</i> and b) With the class <i>Erroneous points</i> . Selection of three classes that have different frequencies in the dataset. Data from PAPER 1.	63
36	Step-wise semantic segmentation for improved differentiation of classes with similar features. With network A, a segmentation is performed for general classes, which is refined in network B.	64
37	Dataset optimization methods for semantic point cloud segmentation: a) Dataset expansion by randomly copying points, b) weighting the <i>loss</i> function, and c) dataset expansion by copying inputs with infrequent points.	65

38	Process for creating an <i>adjacency matrix</i> and applying it as a network input. .	68
39	<i>Eigenvalue</i> based features calculated from geometric features (x, y, z): a) GT semantic segmentation. b) <i>Sum of eigenvalues</i> as feature. c) <i>Planarity</i> as a feature. d) <i>Linearity</i> as a feature. A histogram is shown next to the legend. .	69
40	Semantic point cloud for the classes <i>Objects</i> and <i>Erroneous points</i> . The semantic segmentation is performed using the <i>PointNet</i> -based workflow with the features: x-, y-,z-coordinates, <i>sum of eigenvalues</i> , <i>planarity</i> and <i>linearity</i> . . .	70
41	Semantic point cloud for the classes <i>Building parts</i> and <i>Interior</i> . The semantic segmentation is performed using the <i>PointNet</i> -based workflow with the features: x-, y-, z-coordinates, <i>sum of eigenvalues</i> , <i>planarity</i> and <i>linearity</i>	70
42	Graphical Abstract: Evaluation of semantic segmentation methods using the quality model.	XLI
43	Graphical Abstract: The point clouds are separated into different semantic combinations for the training (first row). Different methods are used to extend the class distribution (second line). A DL algorithm is used to train the combinations and extensions, which are then evaluated according to fixed evaluation criteria.	LXXXIII

List of Tables

1	Commercial tools for semantic segmentation of 3D point clouds, which are not related to a specific scientific work. Abbreviations: Bounding box (BB), offline tool (OT), web service (WS).	21
2	Open-source software for semantic segmentation of 3D point clouds, which are not related to a specific scientific work. Abbreviations: Bounding box (BB), offline tool (OT), web service (WS) and Robot Operating System (ROS). . . .	21
3	Parameters of the hardware and software used for development and testing (single workstation).	60
4	General HPs for CNN architectures. Optimized set of HPs and typical values ranges for these HPs.	61
5	<i>PointNet</i> -specific HPs. Optimized set of HPs and typical values range for these HPs.	61
6	Contribution to Paper No. 1	XXVII
7	Contribution to Paper No. 2	XLI
8	Contribution to Paper No. 3	LXXXIII
9	Contribution to Paper No. 0	CV

1 Introduction

The digitization of everyday life is a trend that has accelerated in recent years, particularly as a result of the global *Corona* pandemic. New ideas on how everyday life and the working life can be digitally designed have been developed and brought to market maturity in a very short time [1]. These applications frequently use data that represents the real world, as an abstract and geometric copy. Creating a geometric model of real-world objects (e.g., building components, structures, countries, continents) is a core competency of surveyors, and has been the basis for maps, three-dimensional (3D) visualizations (e.g., globes), and knowledge [2, 3]. With very fast, precise and easy to use measurement systems for surface recordings, digitization can be preformed much faster, but usually only the geometry and not the semantics is recorded. In this thesis methods are investigated, which allow to generate semantic information from geometric (and sometimes from spectral) measured values. The basis are point clouds, which are recorded with *Light Imaging, Detection and Ranging* (LIDAR) scanners and depth imaging cameras. The point clouds are semantically enhanced by *machine learning* (ML) and *deep learning* (DL) methods. In particular, the constructed environment, i.e., buildings, cities, and long-stretched infrastructure structures, are objects for which semantic segmentation is necessary [4, 5]. The motivation for a reliable semantic segmentation is explained in section 1.1. The two following sections 1.2 and 1.3 explain the *Research Gaps* (RGPs) as well as the research objectives (ROs) and, the research questions (RQs). In the last section of the introduction (section 1.4), the structure of the thesis, the relationships between the sections and the form of presentation are described.

1.1 Motivation

Cadastre, *Geographic Information Systems* (GISs) and *Building Information Models* (BIMs) are the most used applications to represent the real world in an abstract (digital) model and to use them for answering specific issues form topic such as land use, building condition, or mass determinations. These data collections are the basis for public action of administration and economy, strategic planning of social developments and political decisions, so that they are of importance [2, 6].

In the data collections semantic, topological, thematic, spectral and object-inherent characteristics are combined with geometric and geographic object characteristics. Traditionally, these data collections are organized in two-dimensional (2D) representations (e.g., maps or images) in combination with registers (e.g., property registers, land charge registers or land registers). With the advantage of GIS and digital user tools, a paradigm shift has occurred towards the direct storage of object-related information in the form of attributes of a model [6]. A BIM is a data model that represents, among other characteristics, in particular the geometric characteristics of objects as a volumetric 3D model. The great advantage of a BIM

is that the details of geometric characteristics and object information can be represented in a scalable and hierarchical manner. In a BIM, which originally comes from planning, the dimensions of the building objects become more detailed and semantic information become more accurate as the planning proceeds. At the beginning of the planning it is only known that a room needs a door and approximately on which wall it has to be, the position of the door, its shape and materials become more concrete as the planning advances. This is currently represented by the five *Level of Development* (LoDev), [6, 7]. These LoDev, can be further refined for respective characteristics, such as *Level of accuracy* (LoA), information content or *Degree of Modeling* [8, 9]. BIM properties can also be used after completing the construction of a building, such as for comparing the as-built planning with the *as-is* execution (final survey) [10]. In the operation of a construction, BIM is a data format that can be integrated in construction maintenance programs, contributing to the effective and efficient use of a building or infrastructure [11, 12]. Applications include indoor navigation and improved space utilization [13, 14], as well as building control, repair-planning [15], and emergency exit simulation [16, 17].

If data for a BIM is not available from the planning or does not match the *as-is* status, the data is usually surveyed by total stations, photogrammetric or LIDAR systems [6, 18]. Since most methods scan a surface, this process is commonly called *Scan2BIM*. *Scan2BIM* or more generally *Scan2Model* describes the procedure from the recording to a complete model of the real world in an accuracy and level of detail arising from the application [19]. In the *Scan2Model* process, the point clouds are usually combined with other recordings and, if necessary, the calculation of the point cloud is carried out for photogrammetric systems. In terrestrial laser scanning (TLS), which is currently the standard method for most high-accuracy models, the registration is done with common points in the overlapping areas of the scans [18]. Mobile *Multi Sensor Systems* (MSSs), such as scanner backpacks [20, 21] or vehicle-based systems [22, 23] usually use trajectory to connect individual scans, but may also be supported by common points. In most application, filtering is used to partially remove the mismeasurements. The next step is semantic segmentation. Semantic segmentation can be combined with modeling, if a direct automatic or manual creation of parametric geometries is done [24, 25, 26]. These methods are used in applications to create floor plans [27] or surface models (e.g., meshes or voxels) [28] from the point clouds. These models are usually not transferred back into the data format point cloud, but they form parameterizable geometries, such as cylinders, cubes, planes, lines or circles, which are used to build complex object-oriented models, such as GML [29] CityGML [30], IndoorGML [31] or *Industry Foundation Classes* (IFC) [32]. The methods are usually very specific to an application and require detailed prior knowledge about the data and the task. The most common applications for these methods are the modeling of building structures.

However, in most applications, semantic segmentation and modeling are performed independently. The semantic segmentation of point clouds is the most complicated step of this process chain to automate, as the objects vary in geometric size, shape and the recorded scenes differ significantly [33]. Parameters and thresholds for separation by semantic ob-

jects can be insufficiently defined, which still makes ML and DL most suited methods [34]. The performance of ML and DL varies, depending on the data and the complexity of the class definition according to which the point cloud should be segmented [35, 33]. However, strictly point clouds are imperfect data for semantic segmentations, since the training data are usually only available to a small extent, do not have a homogeneous structure, and are erroneous [35]. These disadvantages of point clouds lead to varying semantic accuracies for different classes, to systematic confusions between classes and to unfavorable foundations for modeling [36].

Overcoming the imperfection and understanding its causes for the case of semantic segmentation with DL for building reconstruction is the motivation of this thesis. The aspect of processing point clouds with DL, the quality of point clouds and the generation of training data from point clouds must be examined in a structured manner as it is explained in [37, 38, 39, 40, 41, 42, 43]. Like [36], this work focuses on the point clouds and its weaknesses, as well as methods to overcome them.

1.2 Research gaps in semantic segmentation of point clouds

DL is the most suitable method for the semantic segmentation of point clouds but it has several downsides and aspects that are less researched. The main researches on DL methods deal with the following aspects:

- Optimization of algorithms and network architectures [44, 45].
- Enhancement of the benchmark datasets collections [40, 46, 47, 48].
- Neighborhood representation for algorithms input [49, 50, 51].
- Automatic transformation of point cloud information into parameter models [35, 52].
- Optimization of manual annotations [53, 54, 55].
- Investigation of the impact of point clouds and its pre-processing for optimal semantic segmentation [36, 56].
- Development of quality models and characteristics for semantic point clouds [18].
- Concatenation of DL with ML [57, 58].

The findings for one research aspect sometimes provide the foundation for the others. This can be seen in the example of the development of the network architecture of *PointNet* [45]. This network architecture enables an efficient and direct processing of larger point cloud scenes (> 1 million points). Now, the data pre-processing of the point cloud format is no longer a primary issue, but the data content is a new issue. The main research in the field of DL methods on point clouds does not address real and practical applications, therefore many relevant and influencing parameters are neglected. This leads to the RGPs addressed in this thesis:

RGP 1: Influence of dataset characteristics, points and point clouds for semantic segmentation.

RGP 2: Development of a heuristic description and evaluation of semantically segmented point clouds.

RGP 3: The development of a workflow for semantic segmentation of point clouds in building modeling processes.

The *research objectives* (ROs) are derived from the RGPs, but do not necessarily address the entire research gap. How the RGPs can be closed is explained in more detail in section 1.3 based on the ROs and the RQs.

1.3 Research objectives and questions

The identified RGPs are in the overlapping field of the disciplines of computer science, mathematics, data science, computer vision, civil engineering, facility management, as well as geodesy and geoinformatics. In order to close these gaps, innovative data models and processing algorithms must be implemented by means of modern high-performance computer systems to address topics arising in the digitization of buildings. Recorded digital datasets of buildings have measurement errors, vary in terms of semantic class sizes, include fine and coarse objects in unstructured and heterogeneous point clouds (Figure 1).

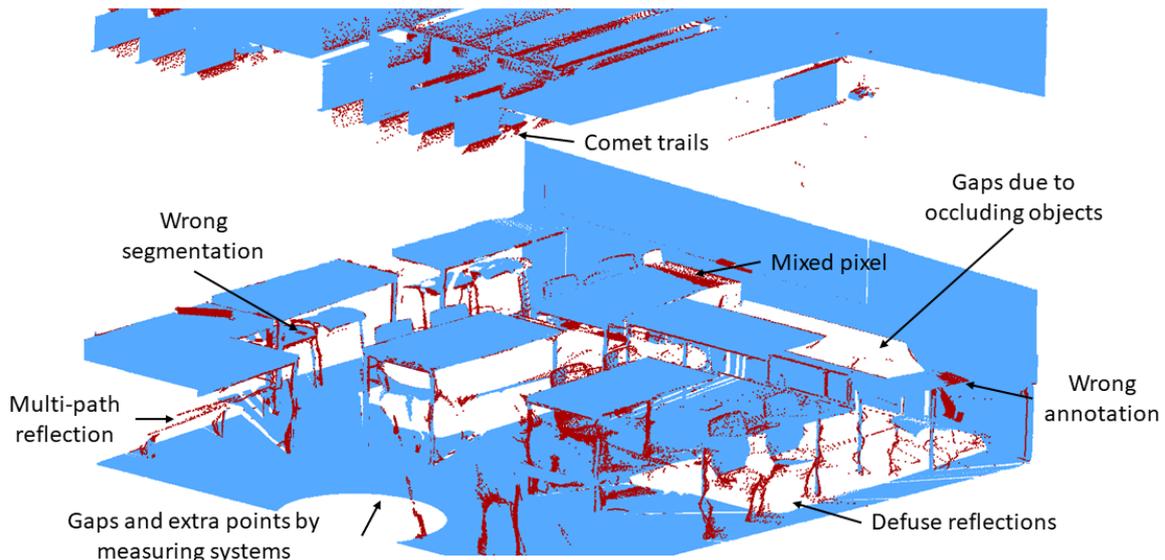


Figure 1: Measured TLS point cloud with segmentation and annotation errors. Class *Errorneous points* in red and class *Object* in blue.

These datasets are not optimal for processing with ML or DL methods, due to the data content and data format. Nevertheless, ML and DL methods are the most efficient and accurate methods for semantic segmentation if the data is homogeneous, structured, and arranged in a raster. In order to harmonize characteristics of point cloud datasets and DL algorithms, the following three ROs are tackled:

RO 1: Evaluation of methods for the manual annotation of point clouds regarding efficiency, usability, accuracy, and the development of an experimental annotation tool.

RO 2: Development of a quality model that heuristically describes semantic point clouds.

RO 3: Development of a workflow for semantic segmentation in order to investigate the influence of point clouds content and format in DL methods.

RO 1 can be achieved by explaining the developing steps of the *Point Cloud Classification Tool* (PCCT) (PAPER 0) and the investigations of its usability. For manually annotated point clouds with a computer, the users must have segmentation tools, a visualization of the point cloud (on a screen), guidelines for the classification process, and tools for the classification. Based on these statements, the following RQs should be answered:

RQ 1.1 Which annotation tools (manual segmentation) for point clouds exist? What functions can be found in these tools? How efficient, reliable and effective are these tools and how can these characteristics be determined?

RQ 1.2 Which annotation tools can be used for the semantic segmentation of challenging real-world indoor TLS point clouds?

RQ 1.3 How can semantic segmentation tools for point clouds be enhanced and improved?

RQ 1.4 How to become a good annotator for semantic point clouds? How can the performance of annotators be measured? What do annotators need and how can the tool support them?

In order to answer the questions of RO 1, a heuristic quality model must be used. The development of a quality model is the RO 2. The quality model evaluates the semantic segmentation process and the semantic point cloud. The development of the model is guided by following RQs:

RQ 2.1 What are suitable semantic point clouds? What are the characteristics of point clouds? How can the characteristics of the point cloud be determined, measured and compared?

RQ 2.2 How is a quality model for semantic point clouds designed? Which parameters are necessary for the description of the characteristics? Does the quality parameters differ for annotation and automatic semantic segmentation?

RQ 2.3 How can the quality model be applied for the semantic segmentations of building point clouds

RO 3 is based on RO 1 and RO 2 and is the realization with the workflow for semantic segmentations. The training data created by the PCCT or other tools and the performance evaluation of the workflow by the quality model are necessary to answer the RQ 3.1 to RQ 3.3. The workflow is developed for the point clouds created by TLS with imperfections as

shown in Figure 1. In the workflow, established DL methods are integrated. The formal and content influencing parameters of point clouds are evaluated in experiments. The RQs which guide the development are:

RQ 3.1 How can DL methods be integrated in a workflow for semantic segmentation of point clouds? Which DL methods are suitable?

RQ 3.2 Which hyperparameters need to be defined for applying *PointNet* in a semantic segmentation workflow? How are the values for these hyperparameters determined?

RQ 3.3 How can the influence of the dataset be controlled by data-based hyperparameters in the semantic segmentation of point clouds? What are the main influences?

The three central ROs are covered by three peer-reviewed and one extend-abstract-peer-reviewed publications. There is no one-to-one assignment of one RO to one publication. Instead, single or multiple RQs are covered in each publication. The connections between the publications and the ROs are explained in section 3.5 and Figure 27.

1.4 Outline of the thesis

This cumulative dissertation consists of a framework thesis (sections 1 to 5) and the four publications in the appendices A (peer-reviewed publications) and B (non-peer-reviewed publication). The framework thesis presents the state of the art, the terminologies (section 2), the connections between the individual publications (Section 3.5), and the ROs, RQs, and results (section 4). Section 5 summarizes the key conclusions and outlines further research approaches.

PAPER 0 is in section B.1 and PAPER 1 to PAPER 3 are in sections A.1 to A.3. A reference to the publications is made by the indication *PAPER #*. References are used to avoid repetitions of results, proofs, and detailed descriptions that have already been discussed in the publications. For better comprehension, conclusions and general observations are discussed in section 3 and in the individual RQs (section 4).

2 State of the art

Semantic point clouds are the foundation for modeling complex environments. Their deployment covers the entire process with the acquisition, the parameter-based filtering and the semantic segmentation of the point cloud (steps 1 to 3 in Figure 2). Based on the semantic point clouds, parametric, solid models, *Computer Aided Design* (CAD) and BIM models are created. These models are used in GIS [59], construction management applications [12, 60], and in private and public registers [61] (steps 4 and 5 in Figure 2).

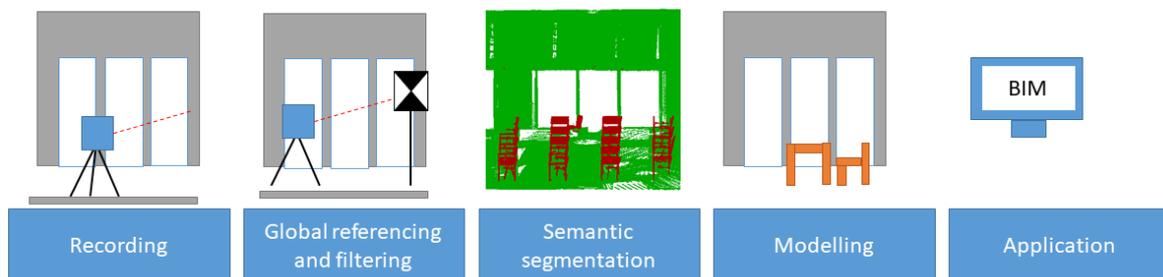


Figure 2: Process of creating semantic point clouds. Recording of point clouds with an optical recording system. Registration of the individual recordings, resulting in a complete point cloud. Semantic segmentation according to given classes set. Modeling of objects in the semantic point cloud. Implementation of the models into an application. Taken from [62] and adapted.

In context of buildings, LIDAR scanners and measurement cameras are used to record entire surfaces in a fast way. The working principle, the differences of the measurement system as well as its influence on the point clouds are explained in section 2.1. The recording and the semantic segmentation are significant for the quality of the semantic point cloud. Semantic segmentations are performed manually as well as automatically and due to the size and complexity of the data this topic comes under *big data*. *big data* applications require special data handling, which is carried out with ML methods. The basics of *big data* and ML are introduced in section 2.2. The process of manual semantic segmentation is in section 2.3. ML models *learn* the relationship between input and target data from the data itself. The characteristics of training data are described in section 2.4. The state of the art of automatic semantic segmentation of point clouds is described by ML and DL methods in sections 2.5 and 2.6.

2.1 Recording systems for point clouds

Point clouds have become a *quasi-standard* for storing recordings and visualizing the surfaces of real objects in the digital domain. This *quasi-standard* can be explained by the fact that many recording systems have to store the measured values very fast (data-stream) and which does not allow an order (sorting) according to the contained object classes [63].

This data organization as a simple list is sufficient since the data is represented as a three-dimensional scatter plot. Humans can interpret scatter plots of point clouds very well and they automatically add semantic content from geometry [64]. For the data recordings itself, different contactless sensor types have become popular and can be distinguished as shown in Figure 3 in active (e.g., LIDAR) and passive (e.g., photogrammetric) sensors.

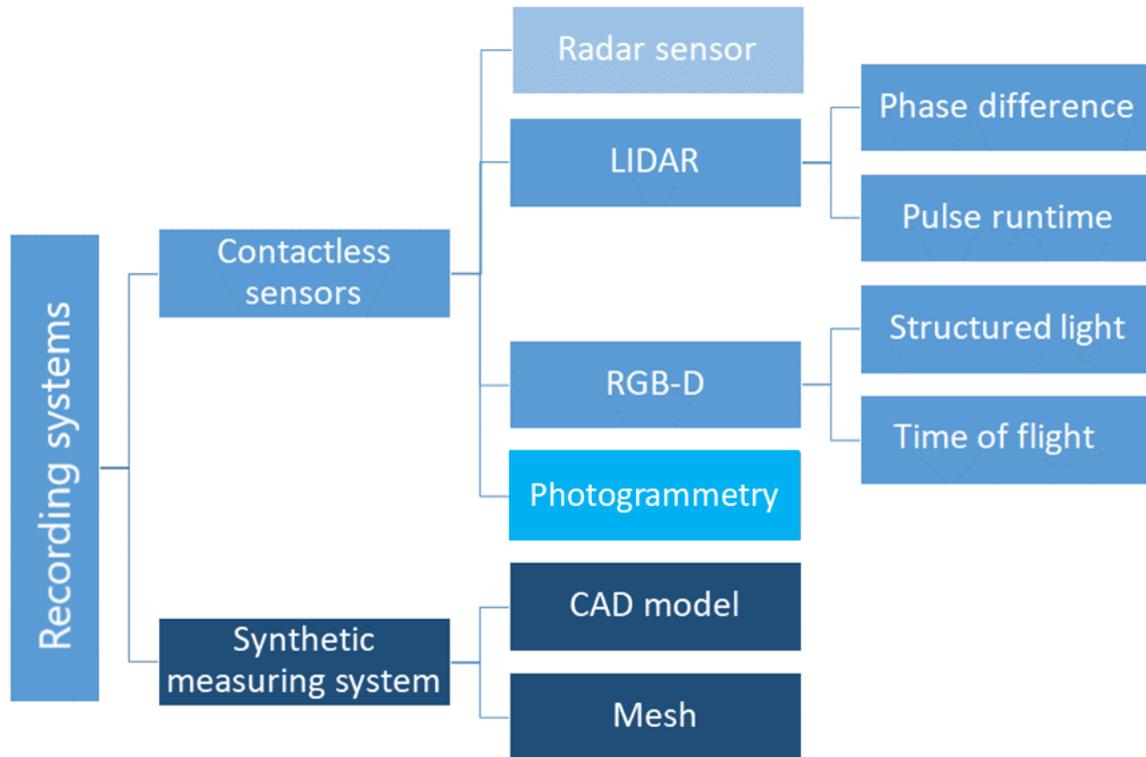


Figure 3: Types of recording systems and the categorization by similar functional principles. Active sensors (radar, LIDAR, RGB-D) send out a signal and determine the distance to the object on reception. Passive sensors such as cameras use multiple images and corresponding points. Systemic data are surface models converted into the point cloud format. Taken from PAPER 2.

Most point clouds of real objects are recorded by optical and contactless recording systems. These contactless systems use either natural light (passive systems) or emitted light from the measurement system (active systems). The light is reflected from the object surfaces and travels back to a photo-sensor in the measuring system, where it is processed. The passive systems use photogrammetric methods to create the 3D point cloud, such as *Structure from Motion* (SfM) or *Dense Image Matching* (DIM). In DIM, corresponding pixels are determined in two or more images. These pixels present the same image feature, for example a sharp corner, an edge or any kind of shapes. The image features are used to determine the alignment of the images to each other (bundle block adjustment). Using the collinearity of the images, a depth map can be created for each pixel in (2D space). By visualizing this depth map in 3D space, a 3D point cloud is created [63]. A detailed outline of the DIM and various developments are described in [65]. Recent enhancements in DIM use DL to improve fea-

ture matching [66]. The DIM method has the advantage that images can be recorded faster and with inexpensive sensor technology. Disadvantages are computationally intensive evaluation, a high noise of the point cloud, a dependence on the surface characteristics and the influence of ambient illumination [67].

With active recording systems, the 3D point cloud is computed directly, no further processing is necessary [68]. Active acquisition systems are depth imaging (Red, Green, Blue and Depth (RGB-D)) cameras as well as LIDAR scanners. Depth imaging cameras record the reflected spectral visible light and the distance to the surrounding objects. *Time of Flight* (ToF) [69] or the *Structured Light* (SL) [70] methods are used. The function of these methods and their differences are presented in PAPER 2. Depth imaging cameras are compact and can be carried freely in space during the recording, so that multiple rooms are (completely) recorded in one measuring loop. The relative easy usability and cheap price of depth imaging cameras lead to a large number of point cloud datasets in indoor applications [71, 72]. Point clouds recorded with these systems have a raster-shaped structure and very low resolution for a single image. With the SL method, multi-pass effects are very rare, but the surface noise is significantly higher for point clouds created with LIDAR scanners or ToF cameras [73, 74, 75] as well as PAPER 2.

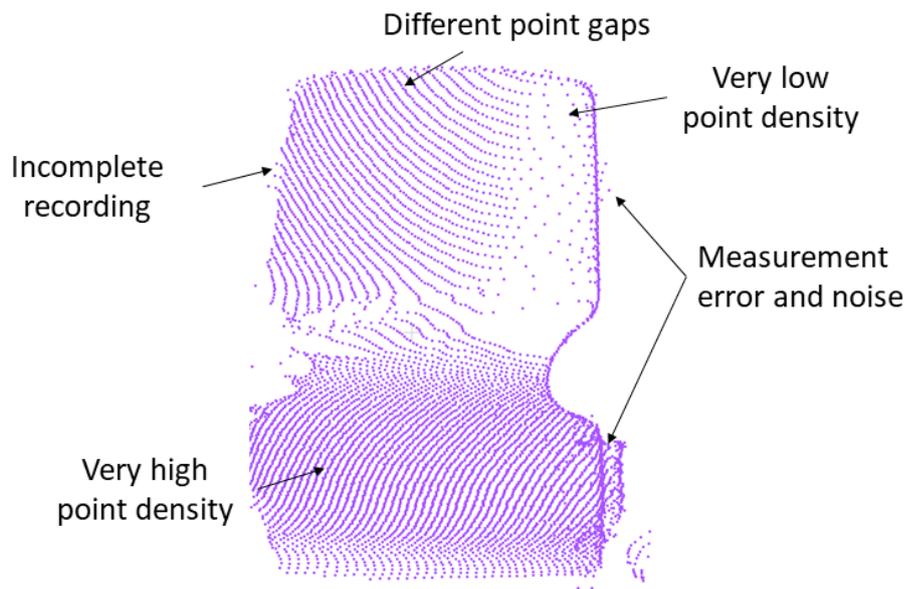


Figure 4: Point cloud of a TLS measurement with measurement errors, noise and non-uniform point density. Taken from [62] and adapted.

For surveying tasks and in outdoor applications, LIDAR scanners are currently preferred and used in the form of TLS [7, 76] or MSS [20, 22, 77, 78]. These LIDAR scanners are nearly independent from ambient light, have a very high acquisition rate (of more than 10^9 points per second). Their range varies between 10 m to several 100 m [79]. Likewise, different distance measurement methods, such as pulse travel time and phase comparison methods, are commonly used, resulting in different patterns and accuracies in the point clouds, as well

as different operational usages. Geometric accuracy evaluation of LIDAR scanners is the subject of many studies in geodesy [80, 81, 82, 83] providing the basis for initial (geometric) evaluation of point clouds. All LIDAR scanners show typical patterns in the point cloud as shown in Figure 4. These patterns include comet tails, multi-pass effects, round-offs at edge, distance and surface dependent densities, acquisition gaps, and phantom points (erroneous measurements). These patterns have a measurable impact on automatic semantic segmentation.

2.2 Big data and machine learning

Big data can provide the basis for generating new knowledge [84]. Typical examples for *big data* are: industrial process data, business data, text data (emails or posts), image and video data, or biomedical data [85]. Also, the 3D point clouds considered in this work are *big data*. 3D point clouds require a high amount of disk space, representing a complex object with different features and sometime with high repetitions rate. *Big data* is fast recorded with little effort and contain hidden information for new knowledge. The knowledge is mostly contained in features of the data and is extracted by *data mining*. A newer term for extracting knowledge from *big data* is *data analysis*, which relates to use computers for processing [85]. Data analysis is done by using statistical, ML or DL methods. These methods are based on models that describe the relationship between features and target data. Target data are either categories (classification) or continuous values (regression) [86].

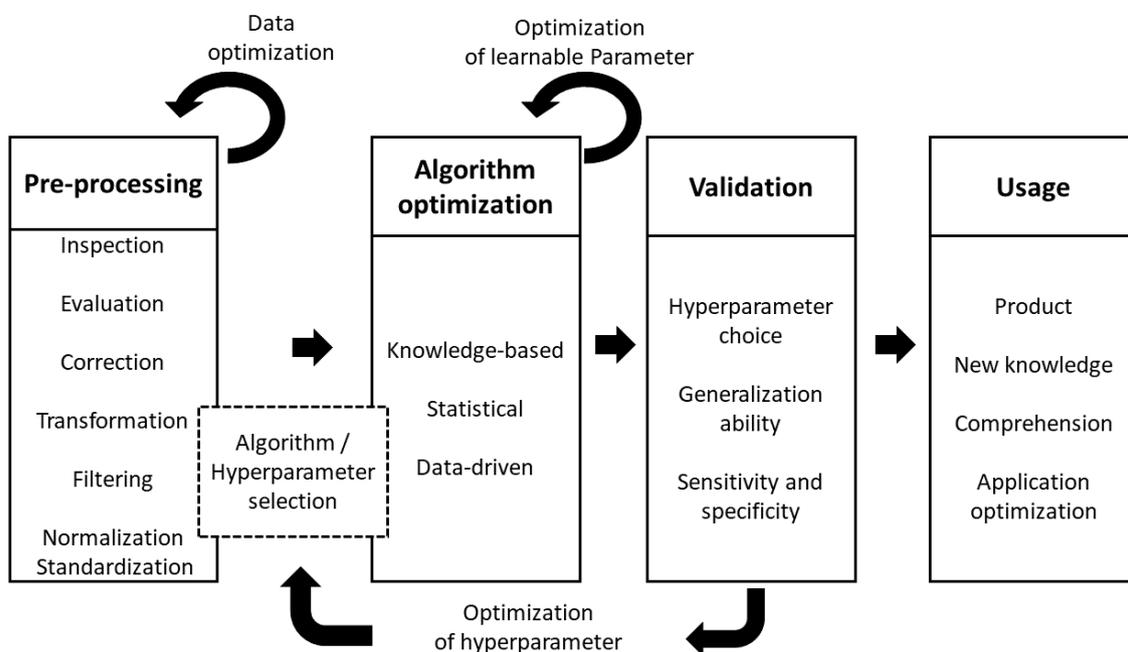


Figure 5: Four stages of the process for analyzing data. Inspired by [85, 86]

In order to analyze data, it must be managed in a data architecture. An overview of different concepts of management, storage and provision of computational resources is given in

[87, 88]. Data features, pre-processing, analysis and evaluation will be discussed in correlation with workflow of data processing shown in Figure 5. This workflow includes data pre-processing, selection and adjustment of algorithms, evaluation of previous steps on the basis of key characteristics of a question and the application of the new data and information. This process iterative in different stages.

2.2.1 Data

The basis of the analysis is the data and its characteristics that are described by meta data (data about the data). The meta data need to be considered prior of any data analysis. By using the meta data it can be investigated if the issue can be solved by this data. If yes it, a algorithm can be selected [89]. The most important meta data are the data format, the datatype, the feature type (variable type), and feature quality. All these terms are frequently confused, depending on the discipline and perspective. To avoid confusion, these important terms are defined below.

Common data formats are images, measurements from sensors, text passages, laboratory results, measured values [86] and point clouds. For each data format exist several file formats that structure the data in a certain manner [89]. The data format images *png*, *jpg* or *tif* can be used. A Point cloud is commonly stored as *pts*, *ply*, *bin* and *las* file formats.

Depending on the data format, the data is binary, qualitative and (discrete or incremental) quantitative. Binary data allows only two states, such as yes and no or *true* and *false*. This data format is often expressed as 0 and 1 for better machine readability and they are discrete quantitative values [90]. Binary values are required for category encoding (e.g., *one-to-hot-encoding*) in ML and DL algorithms. Qualitative data has nominal and ordinal values. Nominal values are categorizations, such as blood groups or semantic classes, that cannot be ranked. Ordinal values are values in a fixed order, e.g. good, satisfactory or sufficient. These values are usually expressed by discrete numbers, e.g., school grades. Quantitative data is data that can theoretically take any real number and usually originate from measurements [91].

The data content, the numbers or categories, are the properties of the dataset and are usually referred to as variables. There are dependent and independent variables in most datasets. The dependent variables will be formed from all or a subset of the independent variables. Thus, dependent variables are aggregated variables that are usually the target variables of a classification. For each variable there are several objects (data points) in the dataset, which have a certain value for each variable. This values are the features of the objects and carry the used information (Figure 6) [89, 90].

The independent variables must describe all the dependent variables as unique as possible. In order to evaluate the suitability of data for processing with certain algorithms, certain concepts have been developed for general data [87] and for traffic data [92], that evaluate the quality of the input data or variables values (features). These concepts focus on character-

		Variables		
		x	y	z
Object	ID			
	1	0,1	2,3	0,2
	2	0,5	1,2	0,1
	3	1,1	0,9	0,4
	4	1,7	2,3	0,2
...				

Features

Figure 6: Definition of variables, objects and features.

istics such as availability, correctness, relevance, completeness, actuality, consistency and comprehensibility. Not all of these characteristics can be used for all data, therefore special concepts have to be developed (PAPER 2). These concepts take the quality of the measurement systems and other pre-processing steps as a foundation. More information on the point clouds recording concepts and investigations for the quality of the recording can be found in [83, 82].

2.2.2 Data pre-processing

The features of the different variables are not always directly usable, because they are distributed differently, have variable value ranges, contain data gaps, contain measurement errors, are highly correlated, are formatted differently, or are too large for direct use. [89] states that in many data analysis projects, 80% of the time is spent on data pre-processing. The data pre-processing depends on the type of data, its recording system and the purpose of the data (analysis) for the development of custom pipelines. Prior to this data pre-processing, an evaluation of the data on possible information errors or limitations should always be performed. This includes checking the suitability of the target classes and the impact of unavoidable dataset bias for the task. The dataset bias is caused due to the fact that a dataset can never contain all possible states. During algorithms optimisation (learning) phase, the method always assumes that all possible states are represented. New data with unknown states are discriminated by the model [89].

In most cases, the first step of data pre-processing is to split the data into meaning groups. These groups can be defined by time or space. The second step is to examine the data for major errors in the features of the individual variables. During this search, data gaps can be detected and, if necessary, closed. In addition, major errors or non-plausible data are

removed or corrected, if possible. The third step is the elimination of random errors, such as measurement noise. This should only be done with filters that do not change the feature appearance [93]. The fourth step is the selection of variables for the algorithms. This can be done by hand (correlation of feature distributions), by algorithms (ensemble learning) or by classifiers [86]. In particular for ML, the selection of variables is crucial for the success of the classification [90, 94]. The fifth step is the transformation of the features into a uniform value range. This can be a conversion of all values into a single unit, as well as the standardization or normalization of the values. This is not always possible, because the variables contain different characteristics of the data. In these cases, the characteristics must be transformed into another feature space. The sixth step is the calculation of new features from the existing ones. Redundant features can be aggregated or target classes can be created. The target classes have to be set up according to the application [89, 93].

2.2.3 Clustering and machine learning

[89] distinguish data analysis for classification and regression. This distinction needs to be further refined. Classifications can be further distinguished to avoid misunderstandings. The different types of classifications are defined in PAPER 2. Following these definitions: Classification is only applied for single objects per data object. The semantic segmentation of the raw point clouds (Figure 7a) is the classification of multiple objects in one scene by semantic type. Usually there are multiple classes. Objects with same semantics are assigned in the same class. Semantic segmentation is understood as forming clusters of objects that describe an object or meaning in the real world (Figure 7c). Clustering is the creation of object groups based on similarities of dependent or independent features (Figure 7b). Instance classification means that not the semantic classes but the objects themselves are distinguished (Figure 7d).

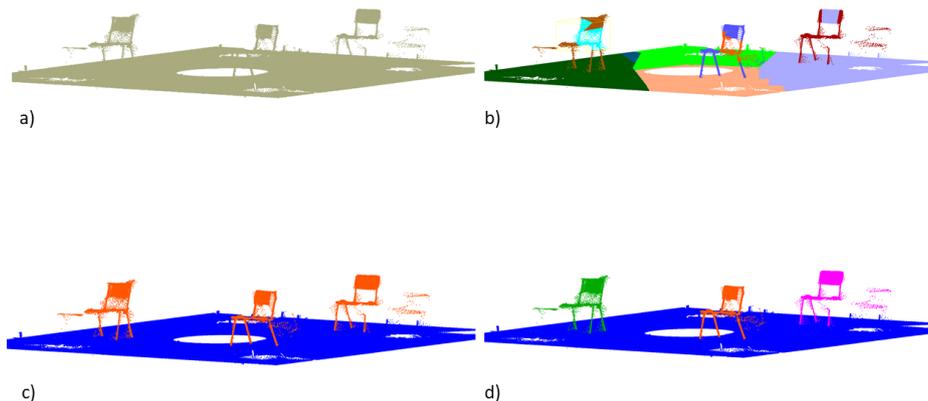


Figure 7: Different classification types: a) raw point cloud, b) clustering, c) semantic segmentation, and d) instance segmentation.

Figure 8 summarizes the frequently described and applied methods for semantic segmentation and clustering of data and organizes them according to knowledge-based and data-

based methods. In addition, a distinction is made between knowledge-based (black) supervised (orange) and unsupervised (green) methods.

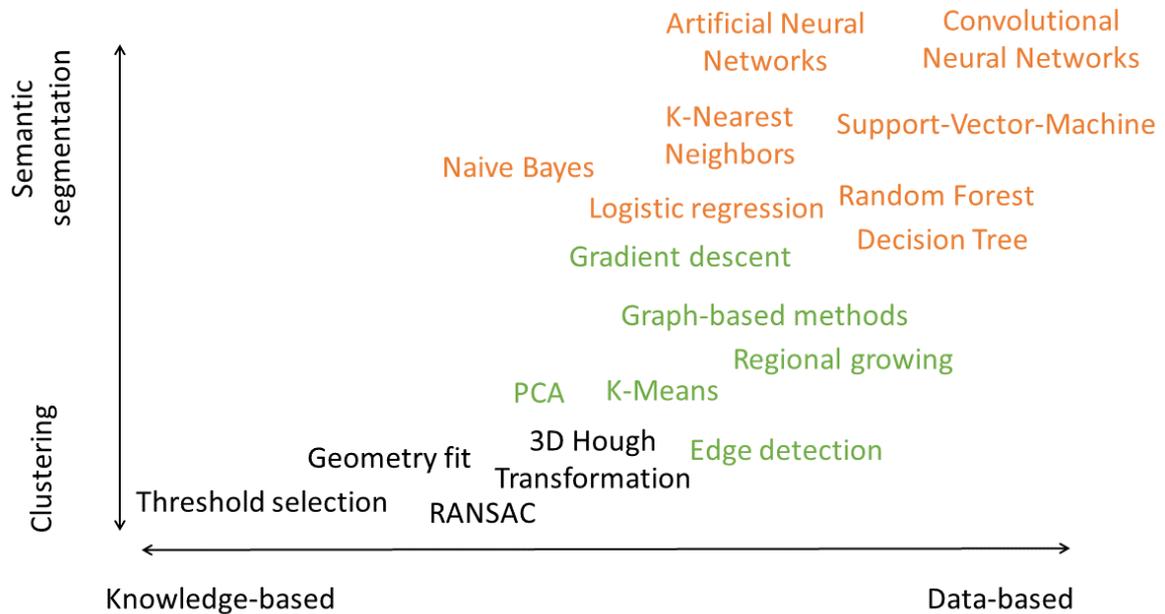


Figure 8: Summary clustering and semantic similarity segmentation methods. Knowledge-based clustering methods (black), data-based clustering methods (green), and data-based semantic segmentation methods (orange).

In programming typically, knowledge is used to process and analyze data (knowledge-based). This requires that the data content is known and can be selected via parameters such as thresholds or number of objects searched. If conditions are met, clusters can be formed using threshold selection [95] (e.g. $\text{threshold} > \text{a spectral value}$) or fitting a geometry in the point cloud. Methods such as *Random Sample Consensus* (RANSAC) [96, 97], require some parameters, such as number of objects and its shape, and randomly search the data for these patterns to form clusters.

In addition to parameter-based methods, which are highly dependent on a-priori knowledge, data-based methods are an alternative that learn the relationship between data and target class from the data itself. Three categories for data-driven approaches are described in the literature. These are *supervised learning*, *unsupervised learning*, and *reinforcement learning*. In supervised learning, the target variables are known and the algorithm learns or determines the relationship between features and the target variables. In unsupervised learning, no target variables are given and a fix or an unspecified number of clusters with high similarity in the features is formed. Reinforcement learning is based on the idea of trial-and-error. The algorithm performs the classification task many times and gets feedback at the end of each pass indicating whether the classification is correct or incorrect [98]. The last method is usually not used for semantic segmentation.

Unsupervised learning is used primarily for clustering data objects. Thereby, differences and similarities in the features are determined e.g. via static method, feature orders or transformation in another feature space [99]. Discriminating methods, such as edge detection [100] or Principal Component Analysis (PCA) [101, 102], define differences by boundaries in the feature space. Based on these boundaries (or thresholds), the unlabeled clusters are formed. Generative unsupervised methods, such as graph-based methods [103, 104], *Regional Growing* (RG) [105], or k-Means [106], start at one or more starting points and grow around the object points that have the greatest similarity. The resulting areas and structures are the unnamed clusters. Using user knowledge or data-based methods, the unnamed clusters become classes.

Supervised learning methods, such as Naive Bayes [89, 107, 108], Logistic Regression [89, 109], *k-Nearest-Neighbors* (kNN) [86, 110, 111], *Support-Vector-Machines* (SVM) [86, 112, 113], and *Decision Trees* (DTs) [108, 114] use target variables to optimize the learnable model parameters. In addition to the learnable parameters, each ML method has additional parameters that must be specified prior to training. These are called *hyperparameters* (HPs) and include the algorithm itself, the proportion of training and test data, *learning rates* (LRs), and stopping criteria. The HPs are discussed in PAPER 3. The previously mentioned methods are mostly classified as weak ML. They allow direct semantic segmentation for a predefined defined set of classes. The labeled data is needed for this purpose (section 2.3). The adjective weak refer to the fact that the features are used directly for semantic segmentation and no depth features are formed from the raw features. This requires that the necessary independent variables have been optimally chosen and that the features are free of gross errors. Data pre-processing has an even larger impact than in DL [86]. To make the methods more robust for varying data, *Ensemble Learning* (EL) methods such as *Random Forest* (RF) [115] were developed. RF use multiple DTs and all are trained under different conditions. The results are combined using methods such as voting, bagging, stacking or boosting. EL also uses different combinations of independent variables to minimize the influence of correlated or irrelevant independent variables. The EL leads to a measurable increase accuracy for most applications [86, 116].

2.2.4 Deep learning

DL methods are usually more robust to errors and major changes in features than ML methods, such as SVM or RF. They have become very important with the rise of *big data*, as they find hidden patterns in large and complex datasets [117]. Commonly, DL is used as a synonym for *Artificial Neural Networks* (ANNs). The functionalities, the different types, as well as the advantages and disadvantages of ANN are briefly explained in this section. The use of ANN for semantic segmentation of point clouds will be discussed in more detail in section 2.6.

An ANN is a mathematical-technical model of a natural neural network such as those found in brains [118]. There are static and dynamic components in the ANN, which are controlled

by the initial HPs. The static components are the processing unit (neuron), the connections (weights) and the network topology (network architecture). The learning phase and the processing phase are the dynamic components [117].

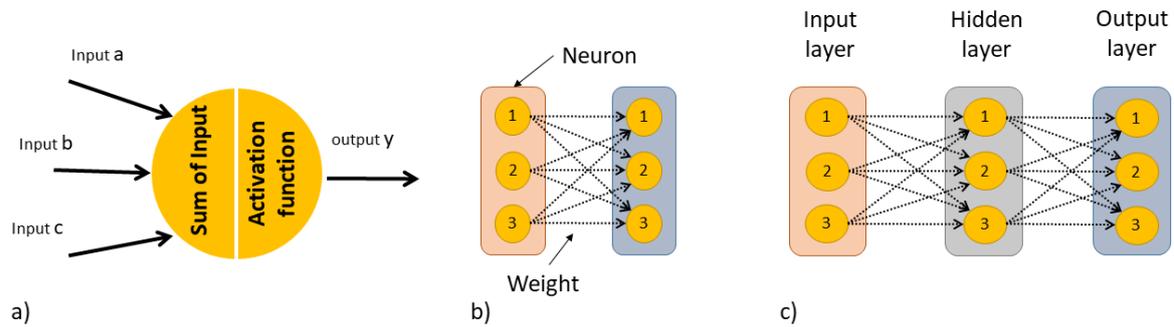


Figure 9: Neuron and network architecture: a) Neuron architecture and function. a) Simple ANN with input and output layers. b) ANN with a hidden layer. Inspired by [119, 120]

The neuron is an independent unit that performs a partial operation of the network. The neuron processes the numerical information by aggregating its input and calculating a new actuation value with a (typically nonlinear) function (Figure 9a). *Step*-, *Sigmoid*- or *ReLU*-functions are used. Neurons are organized in layers and forward parts of the information to other neurons. The simplest ANN consists of only two layers and can only be used for linear problems (Figure 9b) [120, 121]. The input layer has as many neurons as there are independent variables in the dataset and forwards them to the output layer or, in more complex networks (as in Figure 9c), to the hidden layer. In the output layer, there is one neuron for each target variable. The layers are connected by weighted and directed graphs. If all neurons of one layer are connected to all neurons of the next layer, this is called *fully connected* (FC) layer. Also, *sparsely connected* (SC) layers where some neurons have connections are frequently used. Information can flow in all directions. In practice, for static classification tasks, the *feed-forward* (FF) architectures have become most popular. The FF architectures feed information from the input layer through all hidden layers to the output layer. The output layer provides a quasi-probability for each class. A classification function (e.g., *Softmax*) is used to perform the interpretation of the output layer results. During the learning phase, a large number of features along with the labeled class (training data) are fed into the ANN and after each pass, the *loss* is determined across all learning samples. By comparing network predictions and target data, the network *loss* is determined. This loss needs to be minimized by optimizing the weights on the graphs using back-propagation [119, 120]. After the network has been trained several times and the *loss* value is minimized, the ANN can be tested with independent data (section 2.2.5). Once the test parameters are finalized, the ANN can be used in the processing or inference phase [98, 117, 122, 123].

A special type of ANN uses equal weights for all inputs (share weights). These inputs are images or 3D data and carry information in the arrangement of features (neighborhood dependent data). Commonly, ANNs for this kind of data are called *Convolutional Neural Net-*

works (CNNs) [124]. Using CNN, each feature of each variable is loaded as a 1D, 2D or 3D tensor. For each network feed, there are as many tensors as there are variables in the first layer. To extract depth features, each tensor is multiplied by weights of a *feature map* (F-map) and these products are summed up, so that a new depth feature is created from all input variables. The F-map is a tensor, with a fixed width and length (usually a few entries large), that is shifted over the input tensor such that the new features remain local. There are multiple F-maps for each *convolutional layer* (Conv layer), so several feature variables are given to the next layer. The F-maps correspond to the weights in the ANN. When the F-map is longer and wider than one entry, the width and length of the net feature tensor will be reduced (Figure 10). Stronger features are formed by convolution and pooling layers. Commonly, the classification step is done with a FC layer [71, 118, 120, 125].

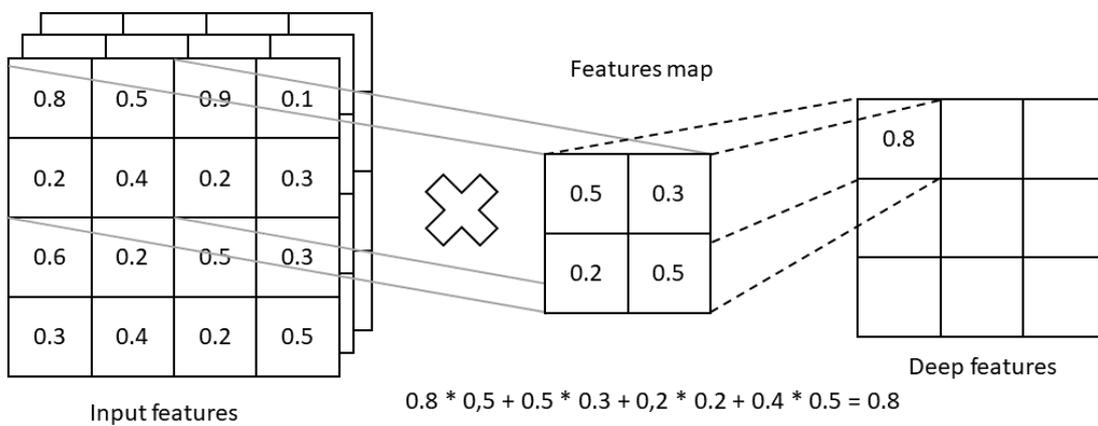


Figure 10: Function of the one Conv layer at a CNN. Inspired by [120].

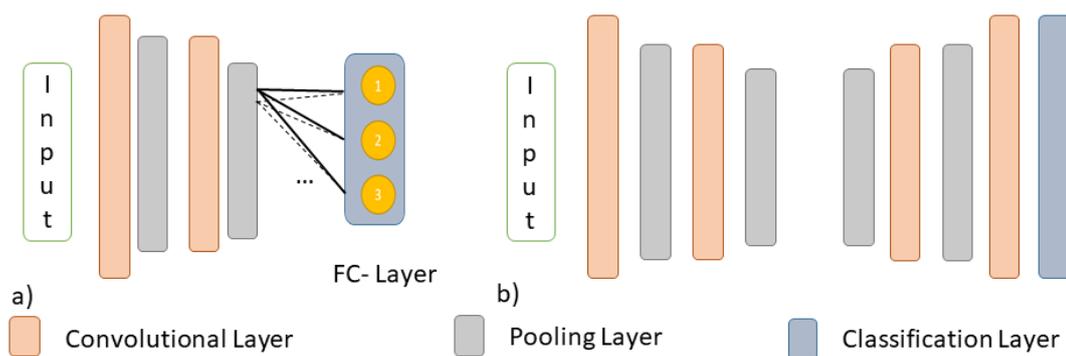


Figure 11: Common CNN-Architectures for semantic segmentation. a) Encoder-Network and b) Encoder-Decoder-Network

Special CNN architectures are the shared *Multi Layer Perceptron* (MLP), *Encoder-Networks* (EN), and *Encoder-Decoder-Network* (EDN). These are often used in semantic segmentations. The MLP is strictly an ANN, such as in Figure 9c, and unit to extract features from data inputs. By implementing this unit with a 1D CNN, more operations with identical weights can be performed in parallel [126]. EN encode the input data to depth features by chained

Conv layer and used at the end a FC layer to classify each point (Figure 11a). This method is used for sparse point clouds or classification questions [127]. In the EDN, the features are encoded and summarized in the encoder phase. In the decoder phase, the features are expanded to the number of input points and decoded (hierarchical approach) [128]. Features can be shared between encoder and decoder layers of the same size through connections (Figure 11b).

An alternative way to distribute information between inputs is to use *Recurrent Neural Networks* (RNNs). RNNs inherit information from previous inputs to the current input and subsequent inputs. The value of the previous information become lower over the time (Figure 12). RNNs are mostly implemented in the form of *Long Term Short Memory* (LSTM) networks, which are explained in [129].

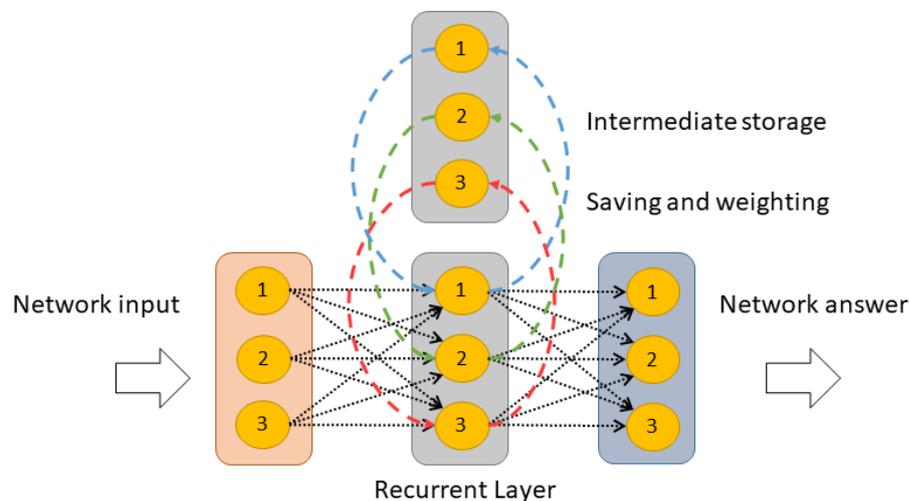


Figure 12: ANN with a recurrent layer. The outputs of the recurrent layer is used as additional input in the next pass. With time the inputs become less meaningful, so that its influence is lowered via weights.

Compared to most other ML methods, ANN and CNN have a high learning capacity, when large training datasets are available. They can be efficiently adapted to new tasks, once the infrastructure for training is set. They usually generalize better than ML methods and are more robust of errors. Disadvantages of DL are long training times, lack of to small traceability of the learnable parameters and there is a need for large amounts of training data [117, 120].

2.2.5 Evaluation scheme and metrics

Most algorithms use intermediate classification results to optimize the learnable parameters, thereby validation is already part of the learning. This validation is done using only very few metrics, which mostly describe the semantic accuracy. Typically, in supervised learning, *Overall Accuracy* (OA) and *loss* are used. In reinforcement learning, binary answers (false or true) are given. In non-supervised learning, no validation occurs during learning in this

sense [89]. The validation during learning gives insufficient information to evaluate the performance of the trained model on new similar data and for each individual semantic class. Before a model can be productively applied, a full evaluation of the model with unknown data must be performed. This must provide information on semantic sensitivity (*recall*) and specificity (*precision*), evaluate the choice of HPs, and provide other metrics such as geometric accuracy [108].

The basis for the evaluation is a *ground truth* (GT) dataset that is used to validate whether the classification for each data point is correct. If this is the case, the point is considered to be *true positive* (TP), if not, the point is considered as *false positive* (FP) in the predicted class and as *false negative* (FN) in the true class. This classification of points is usually presented in a *confusion matrix* [90] (Figure 13) and is the basis for computing other semantic metrics, which [108] describes in general terms. A review of metrics in point clouds is done in PAPER 2.

		True class		
		Class A	Class B	Class C
Prediction	Class A	TP class A FN class B FP class A	FN class B FP class A	FN class C FP class A
	Class B	FN class A FP class B	TP class B	FN class C FP class B
	Class C	FN class A FP class C	FN class B FP class C	TP class C

Figure 13: Confusion matrix for the example of three classes. TP = *true positive*, FP = *false positive* and FN = *false negative*. TP of the one classes is equal to *true negative* (TN) for all other classes.

The automatic classification methods have a large number of HP that have to be customized. The correct choice of HP is the prerequisite for optimizing the learnable parameters and succeeding in classification. In a broader sense, the training of the model is not complete after training of the learnable parameters. Rather, this is only one pass of the integrative optimization of the HPs. This optimization with various manual and automatic methods is presented for general models in [90], for ML methods in [86, 108], and for DL methods in [130]. In PAPER 3, DL methods are reviewed in detail. Evaluation using non-semantic metrics is necessary for special (e.g., geodetic) issues, but is rarely presented in the literature.

2.3 Manual semantic segmentation for point clouds

Semantic point clouds are the basis for creating surface models [4], developing BIM applications [131], building the navigation basis for autonomous vehicles [47], and developing algorithms for automatic semantic analysis of 3D point clouds [39, 132]. Manually enhancing point clouds by segmenting the individual objects in the point cloud and assigning a label is named as point cloud annotation. Point cloud annotation is a very complex task that is time consuming and most often performed by experts [133]. To speed up this task and allow less experienced annotators (e.g., crowd workers) to do this, various software tools have been developed to make annotations more reliable and simple. A brief summary of these tools and providers of these services (*Data as Service*) is given in the following. These tools and their described functionalities form the basis for the PCCT. The motivation of the PCCT is to produce independent, reliable, fast and without additional costs test data for examinations, as there were only few similar tools available at the beginning of this thesis (section 3.1).

The literature review on various manual (open-source and commercial) tools for semantic segmentation of 3D point clouds shows that eight properties of the tools are relevant. These properties are visualization of the point cloud, big-data-capability, tools for segmentation, multi-user capability, adaptability to new circumstances, feedback capability for annotators, semi-automation, and annotation evaluation. An overview is given in Figure 14. A selection of the reviewed tools for semantic segmentation of point clouds, showing methods diversity, is presented in Tables 1 and 2.

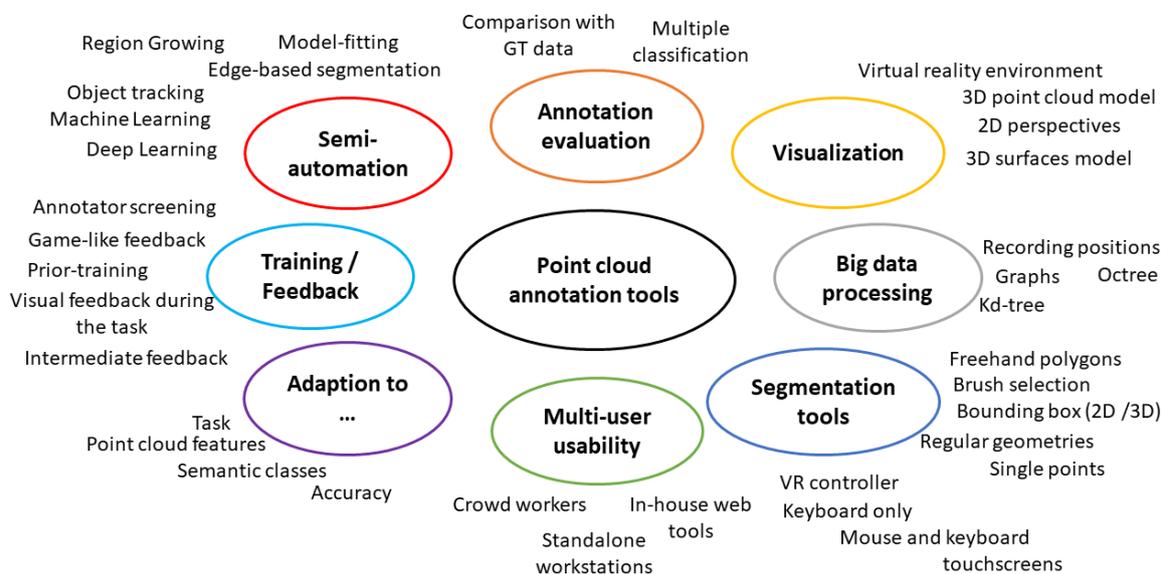


Figure 14: Requirements for a point classification tool.

Table 1: Commercial tools for semantic segmentation of 3D point clouds, which are not related to a specific scientific work. Abbreviations: Bounding box (BB), offline tool (OT), web service (WS).

Tool name	Selection	Application	OT / WS
(AutoCAD) <i>Recap</i> [134]	Freehand, filter, fit, polygon	TLS	OT
PointCab [135]	Freehand, filter	TLS	OT
AWS SageMaker [136]	By own design	All	WS
basic.ia [137]	BB, semi-automatic	Auton. driving	WS,
scale [138]	BB, semi-automatic	Auton. driving	WS
Point Cloud Technology [139]	Data as Service	All	WS

Table 2: Open-source software for semantic segmentation of 3D point clouds, which are not related to a specific scientific work. Abbreviations: Bounding box (BB), offline tool (OT), web service (WS) and Robot Operating System (ROS).

Tool name	Selection	Application	OT / WS
Cloud Compare [140]	Freehand, filter, fit, polygon	All	OT
MeshLab [141]	Freehand, polygon	All appl.	OT
Multi-Label PC [140]	RG	All appl.	OT / ROS
Go Then Tag [142]	Solid fit, pencil	All appl.	OT
PC Annotate [55]	Solid fit	TLS, Auton. driving	OT
SemanticKITTI [53]	Freehand, brush	Auton. driving	OT
3D Annotation [143]	BB, semi-automatic	Auton. driving	OT
LATTE [144]	BB, semi-automatic	Auton. driving	OT
3D BAT [133]	BB, semi-automatic	Auton. driving	WS
SAnE[145]	BB, semi-automatic	Auton. driving	OT

Visualizing 3D data and navigating through it on a two-dimensional screen is described by [146] as a central problem, because the data can only be seen from one perspective, which leads to mistakes in interpretation [147]. [146, 147] address this problem by visualizing the data on a 3D display wall and use a touch screen table for navigation. The idea of processing 3D data in a 3D space is also addressed by the *PointAtMe* application [148], which uses *virtual reality* (VR) glasses for visualization. The annotators wear VR-glasses and can move freely through the point cloud. The annotators segment and classify individual objects via the controllers by placing a *bounding box* (BB) around the points belonging to an object. All other tools from Tables 1 and 2 use a standard 2D screens for the semantic segmentation on which the point cloud is displayed in a predefined perspective [143] or as a free navigable model. The free-perspective choice is default.

The free perspective choice is advantageous for manual segmentation of objects of different sizes. This option allows to look from any angle and at any zoom level at the areas to be processed. However, using this option requires that the point cloud can be loaded in a very high resolution, ideally without delays. Fast loading is an aspect that concerns *big data* capability and is usually implemented by splitting the point cloud into 3D tiles. The 3D tiles are usually realized by *kd-tree* or *octree* methods. These methods organize the point cloud hierarchically, so that only the necessary data section is completely loaded at any given time. The

methods *kd-tree* [149] and *octree* [150] are state of the art in mass data processing [142]. The choice of a hierarchical structure has a great advantage for visualization, because the point cloud can be used in full detail. However, for segmentation, this partitioning can be disadvantageous, because during segmentation the storage structure is changed and has to be recalculated again. In practical applications (e.g., *Recap* [134]), it is observed that these calculations can be reduced if only all points of other classes are deleted from an existing data structure. By deleting the points, the existing data structure remains unchanged and does not need to be recalculated during segmentation. Loading the point clouds with all features into the working memory (direct user access) is very time-consuming, so in many applications only parts of the dataset can be loaded and processed at any given time. The coarse subdivision is usually done according to semantic aspects, such as roads [53], measurement drives [47], recording stations or rooms [27]. Seldom, permanent database systems (e.g., *MariaDB* and *PostgreSQL*) are used for benchmarks, because the data is meant to be exchanged. In addition, folder-based data storage, portable databases such as *5h* or *SQLite* are sometimes used. These formats have the advantage that the data can be loaded via Structured Query Language (SQL) commands efficiently by several users at the same time. Besides solutions for temporary and permanent storage of point cloud data, the filtering of the point cloud according to point cloud density or geometrical aspects is an important aspect. Many manufacturers of recording systems offer optimized parameter-based filters in their own software for point cloud pre-processing or general static filters, such as *Statistical Outlier Removal* (SOR) [151] or *Voxel-Subsampling* and *Fast Cluster Statistical Outlier Removal* (FCSOR) [152]. It is important that the geometry of the object is not changed beyond what has been done by the recording system and that, known measurement errors are minimized.

The annotation of the point cloud consists of segmentation and classification. Traditionally, for segmentation, a perspective is selected in which the object to be classified can be recognized well. The object is separated from the environment with a polygon or lasso and assigned to a semantic class [147]. Besides the free-form polygons or lasso using the mouse, the selection of points is often done by brush technique (sweeping over an area with the mouse) [53], placing BBs over the object [145] or selecting by parametric 3D solids [55]. The selection of points by parametric 3D scenes is done purely by humans, who interpret the 3D scenes differently, vary in the degree of careful work, and are different good trained for the task. This is concluded by [143] under the factor of human error. To minimize this occurring factor, segmentation is often considered as a control screw for automation. The approaches for the automatic geometric segmentation can be summarized in five main methods and one mixed method (hybrid methods). These basic segmentation methods are revisited in section 2.5 and used in a modification for the dataset point cloud. The basic segmentation methods are according to [147]:

- Edge-based segmentation.
- Regional grow.
- Model fitting.
- Traditional Machine Learning.
- Deep Learning.

- Hybrid methods.

For a detailed description of the main methods and examples, references are made in Table 1 by [147]. The list can be extended by the object tracking respectively instance datasets, such as applied in autonomous driving [53, 55, 145].

Multi-user capability plays a minor role in many scientific manual semantic segmentation tools as the datasets are mostly shared by the researchers as in [41, 53, 55]. The annotators mostly process one assigned sub-dataset locally with a specifically developed tool or according to a process description [41, 153] for a general point cloud processing program, such as *Cloud Compare* [154]. Especially when special hardware, as in [148], or extra powerful hardware [134] is used, the scalability by the number of workstations is usually no longer efficient and economical. Multi-user capability is mostly implemented in the scientific context by crowd-working-services (CWS), such as *Amazon-Web-Services*, also known as *Amazon Mechanical Turks* [136]. For example, this is propagated in [133] and considered during software development. Few applications [155] are identified that use the AWS or similar services. Commercial service providers, such as *basic.ia* or *scale*, provide multi-user web applications for various data classification tasks or deliver ready-labeled data. [55] explain in their discussion of the category of annotation tools, that few information is known about the process, the data accuracy and the data privacy.

The commercial annotation services for point clouds focus on the market of autonomous driving. This is done by the BB selection of the data, the initial class sets that primarily include traffic participants, surface types, and street furniture, and the trajectory-optimized visualization and processing. Also, many scientific works, such as [53, 55, 143, 144, 145], are optimized for autonomous driving. However, most of these annotation tools are transferable to mobile mapping applications, because the class selection in these is adaptable or already includes most classes for outdoor applications. Traditionally, indoor point clouds are captured with RGB-D cameras, so semantic segmentation is done with 2D annotation tools, such as *LabelMe* [156] or the tools described in the review by [157]. Point clouds that are sourced by TLS are predominantly annotated using *Cloud Compare* or commercial applications, such as *Recap* [134] and *PointCap* [135]. These tools are optimized for viewpoint-based recording. Each annotation method is usually developed for a specific dataset (data format) and a specific task, and usually requires major effort to adapt to a slightly different application.

The evaluation of manual semantic segmentations and the related feedback and training of annotators are reviewed in detail in PAPER 2. In addition to the statements there, the experiences of [54] can be followed for the training of the annotators. They emphasize the selection of the annotators, the previous experience, an intensive training phase before the proper task and an annotator-bias (individual errors).

2.4 Training data for point cloud applications

ML and DL methods depend on the data they are trained with. Since the creation of datasets is very labor-intensive and cost-expensive (section 2.3), scientific datasets are shared via private websites or repositories [40, 53, 158, 159, 160]. The open-source sharing of large point cloud datasets support many algorithm and application developments. Many of the datasets are related to a specific application (e.g., projects in archaeology [161]) or the development of a new semantic segmentation method [77]). However, this is only advantageous for applications where a large number of datasets is available. For example, many datasets are available for autonomous driving [53, 159] and can only be used for this applications or rated use case, due to the class sets and segmentation type (e.g., BB). In principle, these point clouds (with a different semantic enhancement) could also be used for BIM or city models.

Point cloud datasets have a variety of characteristics, summarized in Figure 15. This set of characteristics are create from [72, 162, 163]. PAPER 2 describes a selection of datasets and outlines the characteristics of recording systems, meta data, semantic segmentation, workflow, applications validation and purpose. These characteristics are usually incompletely described and there is no standard for meta data of training data and benchmarks. This fact limits the usability and the comparability. Among others, the datasets *SemanticKITTI* [53], *Semantic3D.net* [40], *TUM-MLS-2016* [153], are documented almost completely and describe the creation in detail.

Recording Systems	Meta data	Semantic segmentation	Workflow	Application	Validation	Purpose
TLS	Size	Class names	Computation	Method development	None	Evaluation
Mobile	Data	Class order	Pre-processing	Autonomous driving	Multiple classification	Application development
LIDAR	Content	Class definition	Tool used	Robotic	Comparison with GT	Method validation
RGB-D	Provider	Instance description	Procedure	Building model	Samples	Basic research
Synthetic	Data types		Trainings and controls	City models		
DIM	Organization			Forest and agriculture		
	Actuality					
	Usability					

Figure 15: Characteristics of semantic point cloud datasets with main attributes.

Datasets of synthetic data derived from models become more and more popular. Initially synthetic datasets were mostly limited to single objects [164], but in recent years the creation of point cloud datasets for complex scene for autonomous driving [165], outdoor scenes [166] and building modeling [131, 167] have been developed. These data can easily and automatically annotated, but can not represent real measurement errors, gaps and other influences for automatic semantic segmentation.

2.5 Machine learning methods for point clouds

Semantic segmentation of point clouds is not always realized by using the complex and computational expensive DL methods. Alternative methods cluster points by similar features, form boundaries based on a-priori knowledge or boundaries detectable in the data, fit geometric primitives or create surface models, or use ML-based clustering and semantic segmentation methods. In addition, in applications the processing can be directly integrated into the extended measurement process by using overlapping point clouds [168]. Advantages of this method is low computational costs, a low entry level (knowledge and technical equipment), and a high acceptance and traceability of the results.

A categorization of the methods mentioned above can be done according to [72, 169, 170] by basic types as shown in Figure 16. These basic types include clustering, classification, and semantic segmentation methods and will be briefly explained using some examples. In many cases, these methods are combined to efficiently and accurately form semantic segments in a point cloud. [171] combine two clustering methods and use a DT for classification. Combinations of cluster- and DL methods are also applied by [172]. Statistical and ML methods are often used for data pre-processing or post-processing. Typically methods are kNN or K-means as well as calculate the variances.

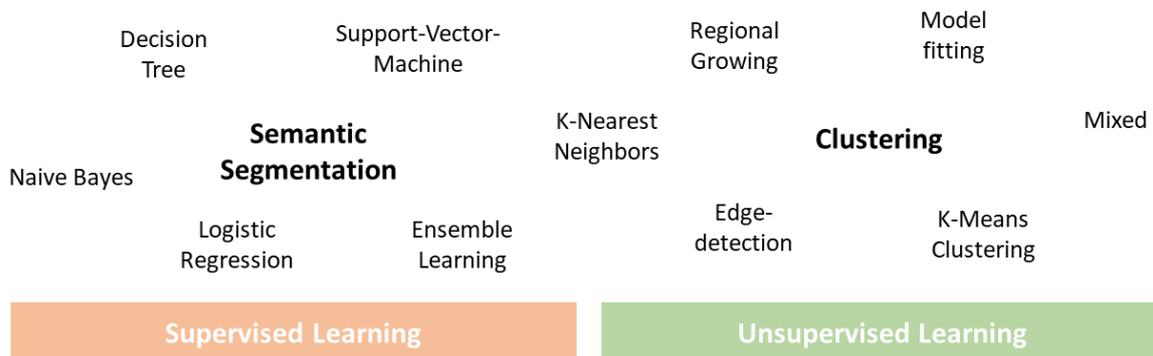


Figure 16: Methods for clustering and semantic segmentation. Left: Methods from ML for direct semantic segmentation. Right: Methods for clustering based on feature similarities.

Edge-detection methods are predominantly used for 2D images, because 3D segments usually cannot close [173]. A projection of the point cloud on a plane is a necessary pre-processing step. For the projection panoramas and *Bird's Eye View* (BEV) 2D representations. BEV are commonly used in navigation applications, for autonomous driving or for the creation of floor plans. Threshold functions applied on the 2D representations can be used to detect feature boundaries, such as Red, Green, Blue (RGB) changes, and make separations within the data. These boundaries can be used to create segments, which can be transferred back to the 3D space or to the point cloud in a subsequent step [27].

The original variant of RG method requires one or more starting points, from which a cluster is formed based on similar features, such as the local neighborhood. In the case if a threshold for the internal homogeneity of the cluster or a too large difference of neighboring data points is detected, the growth of the clusters stop [174]. Besides this bottom-up approach, there is a top-down method where an entire cluster is divided into sub-clusters. RG is usually used in combination with other clustering and semantic segmentation methods. Rasters, voxels, graphs or sub-segments are the most common input formats for RG. Also, clusters can be converted to these data formats to downstream other methods. [175] use RG in a two-step process by forming initial elliptical sub-segments that are transformed into a graph structure and clustered as a minimum spanning tree in case a particular threshold is met (Figure 17b). [176] use a graph that grows directly on the point cloud, investigating the optimal edge parameters. In order to separate foreground and background objects, [177] use a growing kNN graph (Figure 17a) and introduce a penalty parameter for background points. [178] take a similar approach and transfer individual scans from a multi-profile laser scanner into a graph connecting adjacent points (Figure 17c). Using edge operators, the graph is transformed into sub-graphs that represent the geometry of individual objects. A common strategy for large point clouds is to convert the points into a voxel structure. Based on the voxel structure, a graph-based segmentation can be performed [103]. An additional strategy is the use of this voxel-segment-structure for a parameter-based classification of the segments [179].

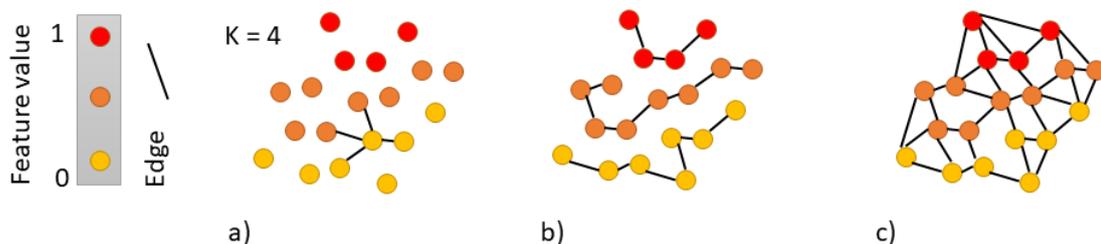


Figure 17: Different graphs used as intermediate step for semantic segmentations. a) kNN-graph with $k = 4$ nearest geometric neighbors. b) Minimum spanning tree by feature value. c) Graph with all adjacent connections for each point. Color of the point indicates the feature value.

The most popular model-based methods for segmentation are the *Hough Transformation* (HT) [180] and the RANSAC method [97]. Originally, the HT is a 2D method to detect circles and lines in images. These planar geometries are used to create semantic segments. In HT, the cartesian coordinates of data points are projected into a HT space, which consists of grid cells and can have different geometric projection surfaces. Using a voting procedure per grid cell in the HT space, the most likely shape and position of the geometries are detected. An overview of the different variants of HT, and the enhancement for a plane detection in 3D point clouds is presented in [181]. Nowadays, the RANSAC algorithm in the version of [96] is a method in which simple primitives, such as planes, spheres or cylinders, are randomly

fitted into the point cloud. A *best-fit* transformation between point cloud and primitive is applied. To increase efficiency, the parameters and the number of primitives must be known. The advantage of these two model-based methods is that already parameterizable models are obtained in addition to the segments. Parameterizable models are the desired output products, especially for building and construction modeling [182].

The following methods are closely classified as ML and include DT and RF algorithms. Infrequently, SVM and *logistic regression* are used in the context of point clouds. ML methods are well established to produce reliable and accurate semantic segmentation results when the HPs are known and correct features are selected. Several papers [39, 41, 94, 183, 184, 185, 186] explain the significant HPs and explore the features variables in detail. A workflow to consider all these influences is shown in Figure 18. These influences are neighborhood selection, feature extrusion, and feature selection. Strategies to determine the optimal neighborhood (size or number of points) in the case of heterogeneous point cloud densities are developed and experimented. For this purpose, the *eigenvalue*-based features flatness, curvature, or entropy are usually computed and used for sub point clouds [39]. Besides measured features, such as RGB values, intensities, and coordinates, mix features and features concerning the local distribution (e.g., *eigenvalues* or point densities) are used [187]. Using methods such as PCA, the dominant, uncorrelated, and significant features are determined in multidimensional feature space. This is necessary because the semantic segmentation performance can be lowered by correlated features and the computations get unnecessarily complex. The author of this work believes that there is great potential for transferring this approach to DL application, even though DL methods are not directly dependent on feature selection. Approaches to this research and initial findings are discussed in section 5.2.

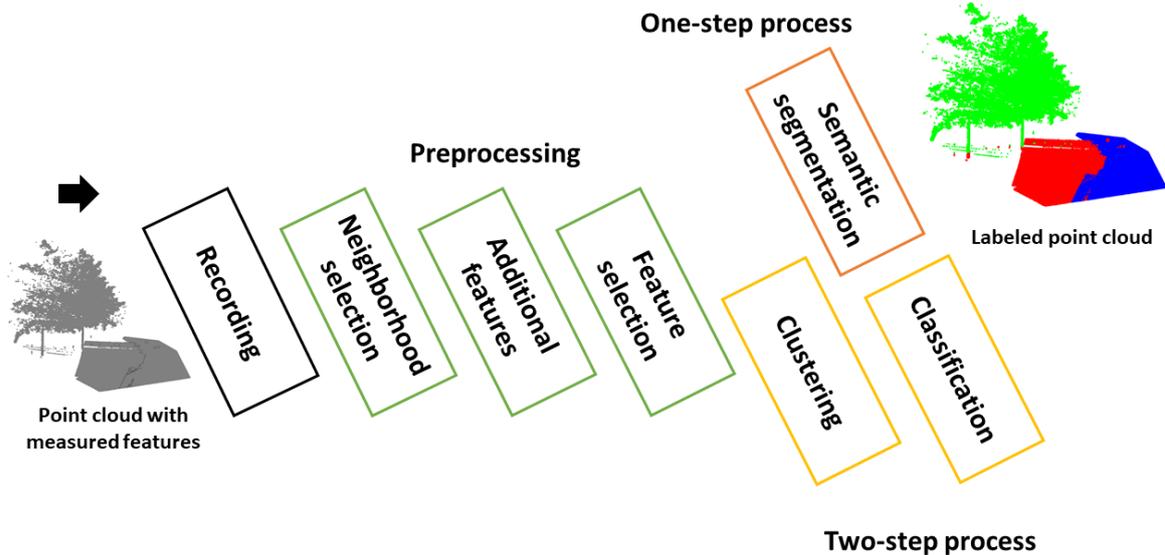


Figure 18: Workflow for semantic segmentation with ML. For non-DL methods, the initial features are important for the success of the semantic segmentation method (green boxes). In unsupervised methods, clustering and classification are separated commonly in two steps (yellow boxes). In supervised learning, it is usually done in one step (orange box). Inspired by [94].

Data transformation, feature analysis, and various clustering and classification methods are concatenated in complex workflows (Figure 18) and benchmarked against DL methods. Comparisons show slightly minor semantic accuracies in most applications [188]. RG for coarse segmentation refined with RF are combined using indoor point clouds as an example in the work of [189]. For the reconstruction of historical buildings [190, 191, 192] apply the above discussed methods and ideas in workflows for semantic segmentations with a RF.

2.6 Deep learning methods for point clouds

In the semantic segmentation of image content, DL networks show a performance that surpasses traditional ML methods [188]. This leads to the enhancement of models for the three-dimensional domain and to the development of different approaches for the semantic segmentation of point clouds. An overview of the central DL developments of the past ten years is presented in the following¹, by a categorization of network types and the method pipelines on the basis of the input data formats. The input data format is the format in which the point cloud fed into the DL network architecture. In this context, projections into 2D domain are *early approaches*², which were followed³ by 3D structured DL methods. Recently⁴, point-based approaches have been intensively researched and are the most popular ones when high accuracy is required. This coarse method categorization can be further refined by sub-types, as shown in Figure 19.

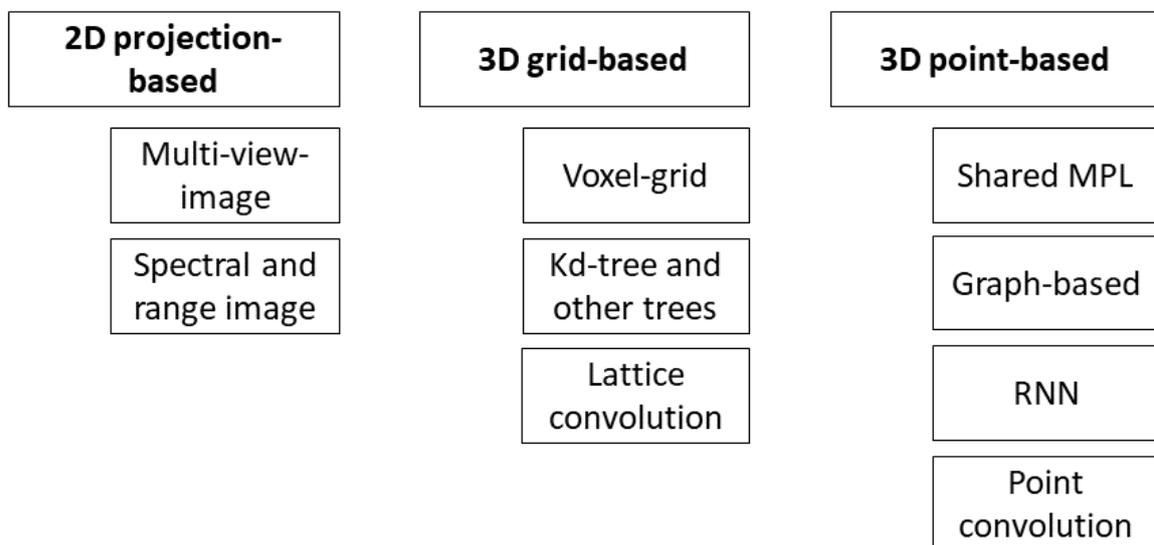


Figure 19: Method characterization of DL network architectures for semantic segmentations by input formats and applied architectures.

¹The research questions of this work base on these methods.

²From 2014 to 2016

³Since 2015

⁴Since 2017

In addition to the methods that can usually be categorized clearly, there are many combinations of approaches. For example, [193] first transform the point clouds into cylindrical coordinates, convert them into a voxel grid and processes them by a FC network. Finally, a point-wise classification is performed. A voxel structure transformation is performed by *VolMap* approach, which transforms the point cloud into BEV perspective, making the geometric height a feature value. The processing is done with 2D CNNs and the pixel labels are passed via the voxels to the point cloud [194]. In *LatticeNet* [195] and *VV-Net* [196], the grid-based and point-based architectures are intertwined to discover local and global features. In *LatticeNet*, the point cloud is firstly converted into the form of a lattice and processed using an encoder-decoder architecture in which *PointNet* layers exist [195]. *VV-Net* focuses on the local relationships within a voxel cell, which is encoded by a sub-CNN to provide depth features as input for a group-based 3D CNN. Finally, the semantic segmentation is performed using a *PointNet* layer [196]. In the following, the general characteristics of the DL network architectures are presented at the beginning of each section followed by specific modifications and extensions.

2.6.1 Semantic segmentation with 2D projection-based deep learning methods

A 2D projection-based method transforms a point cloud with 3D geometric variables (x, y, z) into a geometric 2D format with the variables x and y . The spectral features do not change by this transformation, but the reduced geometric variable (e.g., z) can become a spectral or radiometric variable. The transformation usually results in representing the point cloud as an image. An image consists of discrete pixels and has a fixed *width* and *height*. In the following, variables x and y become *width* and *height* (BEV case). The transformation from point cloud to image transforms the points into a discrete format (pixels), so that a neighborhood relationship can be built through the pixels. With the transformation, the features of the points are passed into pixels, thus it is possible that several points describe the content of one pixel. Due to the transformation, the content of the point cloud is generalized, which is a central weakness of projection-based methods (Figure 20). Points that occur in different ranges, but are at the same location, are combined this way. Usually these points belong to different semantic classes. The major advantage of transferring point clouds to the image is that the established semantically highly accurate and very fast CNNs for semantic image segmentations such as *AlexNet* [197], *U-Net* [128], *Yolo* [198, 199] and others, can be applied [163, 200].

Applications of autonomous driving and robotics use projection-based methods frequently. In these applications, the point clouds are usually very sparse, very reliable (binary) segmentation is required, and the sensors measure in one fix direction of view, such as BEV or *windshield view* (WV). For BEV applications, methods are developed by [201, 202]. Point clouds in for application purpose are recorded with a profile scanner scanning horizontally the environment of a vehicle. Along the z -axis, the point cloud is mapped into a raster image with pre-defined width, heights, and pixel size. The raster image (Figure 20) is evaluated

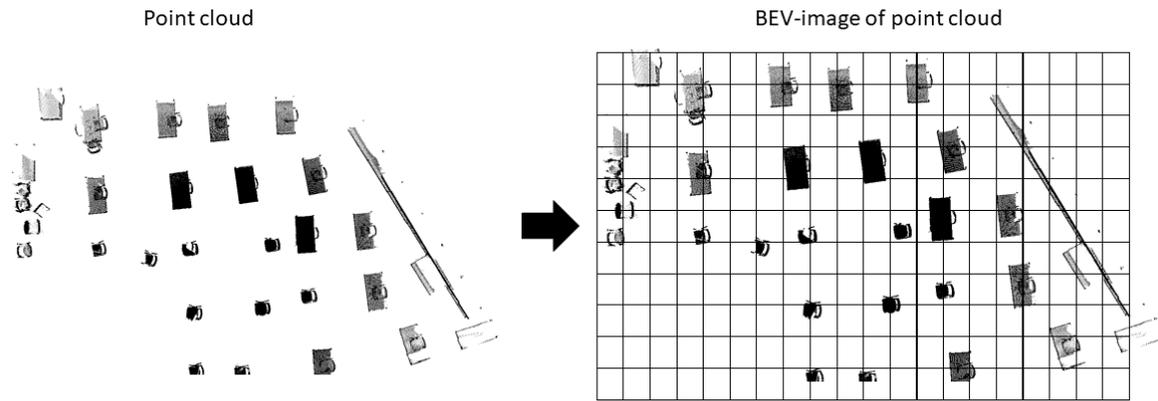


Figure 20: BEV projection. The point cloud is oriented along the z-axis and transformed into a raster plan with fix a fix raster structure.

with a FC network. In the inference phase, the passable area of each scan is semantically segmented in a few microseconds with approximately 90% *recall* and 90% *precision*⁵ [202]. Following this task and approach, [203] have developed a similar method using the *LoDNN* Network [202] for semantic segmentation. There, in addition to the geometric features, features such as pixel densities are used in order to take generalization into account. Furthermore, WV point clouds are semantically segmented with the *U-Net* [128], after they have been transformed into a panoramic image. Different resolutions of the panoramic images are evaluated with the *U-Net* and then the sub semantic segmentations are combined to a joint one using thresholds. Its performance is validated on the *SemanticKITTI* benchmarks [53, 204] and is close to 90% for *recall* and *precision*. *SurfConv* [205] and *PIXOR* [206] aim to detect individual objects, such as other cars, pedestrians or cyclists, which are of a particular interest in the point cloud. This is done in the first steps as described above. The semantic segmentation is performed using a CNN chained by FC layer, where the FC layer is used to express the location, orientation, and reliability of a BB that envelops the object. The accuracy of semantic detection varies between an average *precision* of 55% and 75% [206]. A complete semantic segmentation of WV scenes are intended with the methods *SqueezeSeg* [58], *RangeNet++* [207], *LU-Net* [208] and *SalsaNext* [209]. These methods do not differ fundamentally in the scheme of data processing. In all works, the point clouds are projected onto a sphere, which is then unrolled as a panorama (Figure 21a). Furthermore, different improvements for the geometric resolution are developed and applied. The back projection from the image to the point cloud is addressed and optimized by a kNN step [207] and *Conditional Random Fields* (CRF) [58]. Besides *U-Net*, *Darknet53* [199], *SqueezeNet* [210], and *ResNet-18* [211] are used and adopted. The performance of these methods varies from 52% to 60%⁶ *Intersection over Union* (IoU).

TLS point clouds and point clouds generated with mobile MSS have a much higher density and cannot be mapped from a single perspective. Points would be missed by occlusions

⁵Published by the developers.

⁶Validated on the *SemanticKITTI* dataset by [46].

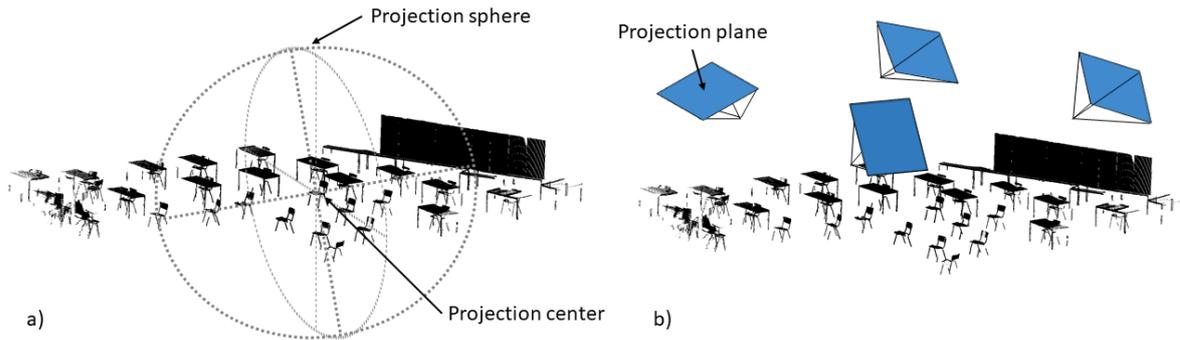


Figure 21: Projection of a 3D point cloud into (2D) image. a) Spherical or cylindrical projection. b) Multi-view-image projection and transformation.

during semantic segmentation or the geometric context would not be identified. Multi-view-image approaches (Figure 21b) are developed by [212, 213]. The point cloud is considered as a surface and a mesh is computed from it. [212] use randomly generated images that represent completely the mesh at different distances and rotations. These images are semantically segmented with EDNs, such as *U-Net* and *SegezeNet*, and the semantics are projected back onto the mesh. From this, the semantic information is transferred to the point cloud. The approach of [213] use planes that tangentially intersect the point cloud in one point. Starting from this point, all neighboring points in small area round that the tangential point are projected into the plane, and the areas without information are completed by interpolation between points (Figure 22). The plane is overlaid with a pixel grid and all images are semantically segmented by *U-Net*. The IoU for these methods varies between 51% and 67%⁷.

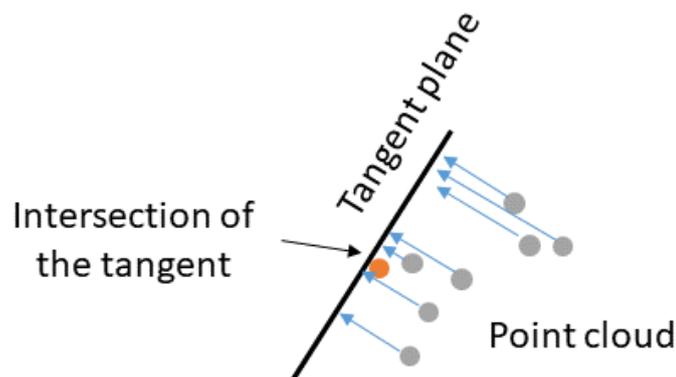


Figure 22: Projection of the points onto a tangent plane. Creation of a multi-view image (Simplified 2D illustration).

A recent work use 2D CNN for semantic segmentation of 3D point clouds by using the recording profiles [214], fuses the point clouds with other data such as images [215], or uses 3D CNN in *air-borne laser scanning* (ALS) analysis [216]. [214] use a profile laser scanner that

⁷Validated by developers on the *Semantic3D.net* dataset [40].

generates a 2D point cloud. This point cloud is transformed into a raster image and evaluated with a 2D CNN. In the data fusion method, [215] use images for the semantic segmentation. A mobile mapping system captures the images synchronously to the laser scans. The semantic information is generated in the images and transformed to the point cloud. This method requires a very accurate synchronization and calibration of the scanner and the cameras. 3D CNN have a wider geometric dimension and convolute the dataset in three directions. This is computationally intensive, so [216] additionally transform the ALS point cloud into orthophotos.

In addition to raster-based 2D CNN, lattice representations of the point clouds are used. The lattice approaches use a bilateral Conv layer [217] to transform the point clouds into the 2D structure. They could be processed with 2D CNN. Advantage of these methods is that 3D points and georeferenced images can be fused, as in *SPATNet* [218], and be used for semantic segmentation.

2.6.2 Semantic segmentation with 3D grid-based deep learning methods

In the context of 3D grid-based DL methods, point clouds are converted into 3D raster structures for evaluation with CNNs. These can be voxel-based, tree-based or lattice-based. This intermediate format allows to apply Conv layers convolut in three dimensions (3D CNN). Most of the 3D CNN architectures are based on the 2D CNN architectures [219] applied to raster images. For semantic segmentation, EDN and EN with a FC layer are used, which assign a class to each voxel. By interpolation in the 3D space, the labels are transferred to the points or further refinements of the semantic segmentation are performed. A general pipeline for these methods is shown in Figure 23. The advantage of using 3D CNN is that the information is preserved in all dimensions and the segments are not mixed in the reduced dimension. A majority of 3D CNN networks commonly merge multiple points into one voxel, so that the data is generalized as well (Figure 24a). The main disadvantage of 3D CNN compared to 2D CNN is that the application and training times are significantly increased, since the number of operations is potentiated. This has led to many early architectures consisting of few layers [124] and the voxel structure being transformed into an occupancy grid (Figure 24b) [220].

The EN with FC layer performs a classification for each network input. In *VoxNet*, the detection of objects in the point cloud is performed with such an architecture [124]. [219] proceed identically, but only label the voxel representation. For the transfer from voxel to point, an intermediate step is introduced that takes into account the distance to the voxel center point. [40, 221] use a sub-voxel grid computed for each point as network input. Small-dimensional Conv layers with 16 x 16 x 16 voxels are used as a basis. In the architecture of [40], the local neighborhood is additionally considered by using voxel grids with five different voxel edge lengths (2.5 cm to 40.0 cm). For each of these five voxel grids, a *VGG16*-like [222] network architecture that is extended as 3D CNN is used. All the sub-CNN results are combined and the classification is performed in the FC layer.

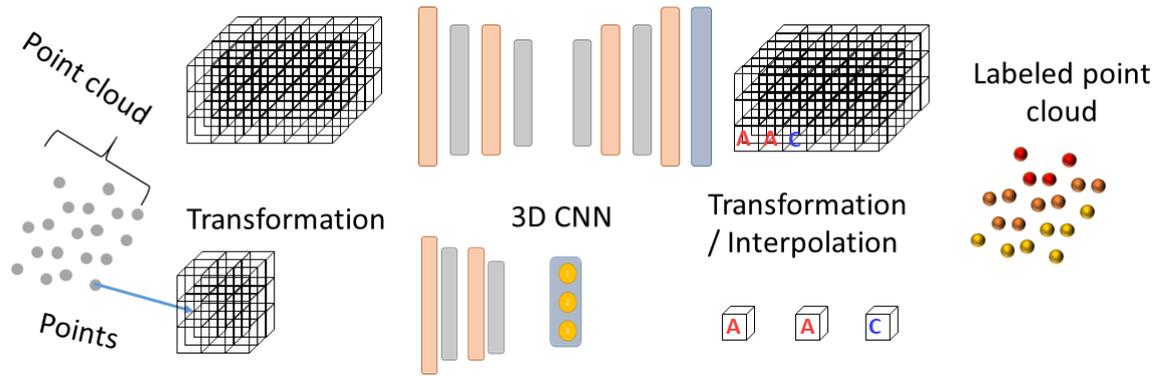


Figure 23: Workflow semantic segmentation utilizing 3D grid structures. Top: The entire point cloud is transferred to one grid. Iteratively, several voxels are fed into a 3D CNN. The voxel grid is semantically segmented. The information is passed by interpolations to the point cloud. Bottom: A sub-voxel grid is created for each point. Each sub-voxel grid is classified by the 3D CNN. Each point is directly assigned to one class.

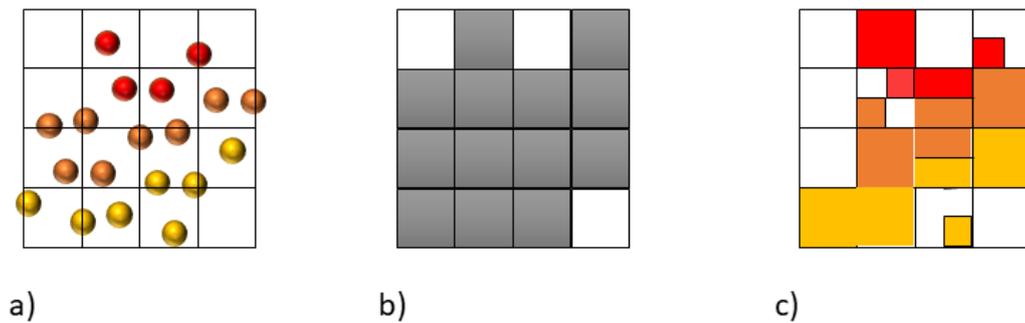


Figure 24: Voxel structures (in 2D perspective). a) Regular voxel grid. b) Occupancy grid. c) Octree with refinement due to point cloud density.

Other 3D CNN are based on a EDN, where a part of a point cloud is transformed into a voxel structure. The FC network can act as an encoder to detect a specific object in the point cloud. *Vote3D* [223] and *Vote3Deep* [224] use such a CNN architecture, embedded with a voting algorithm, to detect the BB of objects relevant to autonomous driving. EDN for spatial classification of RGB-D images find applications in the architectures: *SSCNet* [225], *ScanNet* [226], and *ScanComplete* [227]. The voxel-resolved RGB-D image is given as an occupancy grid with a resolution of several centimeters as input to the CNN. In this case, the key feature is occupancy or non-occupancy state of the voxels (Figure 24b). Through the EDN, the occupancy grid is directly classified and a transmission of the semantic labels is made to a point cloud or a mesh. [227] use a hierarchical-chained architecture to efficiently increase segmentation resolutions. Additionally, this type of architecture can fill gaps created by recording perspectives [225, 227]. The *SEGCloud* architecture has a CRF layer after the 3D CNN layer, which is used for finer (sub-voxel) segmentation [228]. Thus, combining traditional ML and DL methods contributes to an efficient and easily increase in semantic segmentation accuracy.

OctNet [229] uses an octree as voxel structure (Figure 24c). With the octree, the sizes of the voxel cells are adjusted based on the occupancy of the voxels [230]. *OctNet* is used for semantic segmentation of point clouds representing larger facades. In this context, the height of the facade specifies the maximum size of the octree. The features of the points falling in a voxel are combined by computing an average and calling this value voxel feature. The GT label is determined using the dominant class of the points in this voxel. Semantic segmentation is performed using a 3D EDN, so that directly the selected portion can be semantically segmented. Besides octrees, kd-trees [231] are also used for small point clouds applying FC layer or 1D-convolution for feature extraction and semantic segmentation [232].

2.6.3 Semantic segmentation with 3D point-based deep learning methods

The DL methods from sections 2.6.1 and 2.6.2 have the disadvantage that points always have to be converted into a raster geometry and information is generalized. Direct point-based semantic segmentation became popular with the development of *PointNet* [45]. *PointNet* uses MLPs to extract depth features, which extracted individual for each point. The individual point features are combined via a *max-pooling function*, resulting in *global* features describing the dominant features of all points currently fed into the network. A detailed description of *PointNet* can be found in PAPER 1 and PAPER 3. Due to the point-wise extraction, the order of the points is not important. However, this also has the disadvantage that neighborly relations, which are described by several points, are not considered in semantic segmentation. Moreover, the *global* features refer only to the current input, consequently in most cases the features describe only a very small part of the point clouds. In principle, semantic segmentation can be performed using point-based features, thus researchers use *PointNet* or parts of *PointNet* frequently. Besides *PointNet*-based enhancements, which will be discussed in more detail below, *RandLANet* [44] is one more recent developed network for semantic segmentation of large point clouds. In *RandLANet*, the point clouds are semantically segmented in one step by randomly reducing them.

The key weaknesses of *PointNet* concerning neighborhoods are directly addressed in [233, 234, 235, 236, 237, 238] but non of these works overcome all the weaknesses. In *PointNet++* [233], *PointNet* layers are integrated in an EDN. This network considers the hierarchical local neighborhoods of the points. The central modules of *PointNet++* are sampling, grouping and feature extraction with *PointNet*. The *Farthest Point Sampling (FPS)* algorithm is used to detect principal points. The features of the near surrounding points are grouped in every principal point and fed as one unit to the *PointNet* layer (Figure 25). With these extensions, larger point clouds can be segmented semantically. The same objective but with two different approaches are pursued in [234]. Their first approach uses different sized input levels (area sizes as shown in Figure 26b) and combines features of them. The features generated from the input levels are given in a consolidation unit. The second approach uses a fixed input block size (Figure 26a). The features from different input blocks are fed into a RNN consolidation unit. The information from four input blocks are considered as information

sequence and through the RNN, shared features are created that are used for the semantic segmentation [234].

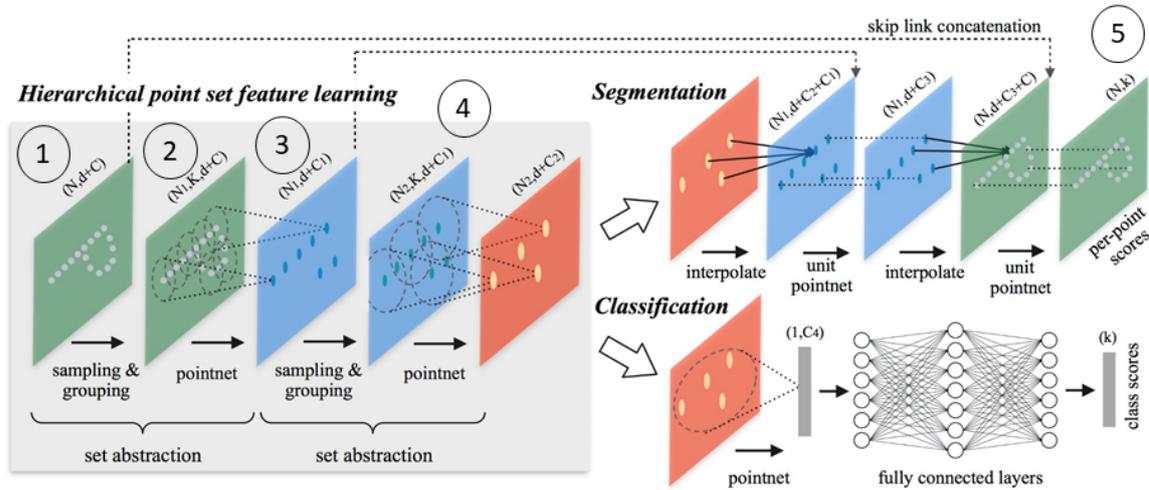


Figure 25: Structure and functions *PointNet++*. Encoding of point features in an iterative process considering the local neighborhood: (1) Selecting n points that are maximal wide away from each other. (2) Grouping of the features in the neighborhoods. (3) Applying a *PointNet* layer to feature extraction. (4) Repeating this process. Decoding by joint and step-wise interpolation of the features. (5) Classification layer at the end. Figure from [233].

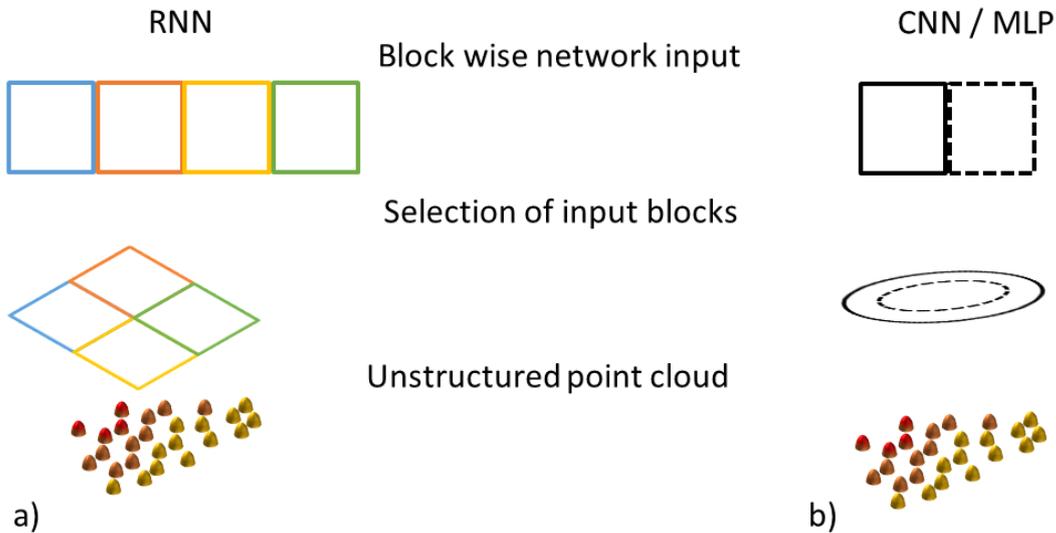


Figure 26: Two methods for creating neighborhood input blocks. a) Fixed block size with the blocks sharing features via RNN. b) Variable block size with different fixed or dynamic radii. Used in CNN or MLP architectures. Inspired by [233, 234].

Similar enhancement is described in [50]. Local multi-scale neighborhoods are implemented by a pyramid pooling function and information is distributed across the network via a RNN layer. This enables learning from the coherence of objects. The enhancements of input features and introducing Conv layer instead of MLP are described in [236]. *Self-organizing*

(SO) networks that use SO maps as inputs and *PointNet* as encoders are developed in [235]. A feature network consisting of concatenated feature modules and a *PointNet* network was developed by [237]. In the feature module, local features are formed by the kNN and global feature by the K-means. Furthermore, a centroided *loss* feature is introduced. [238] use the *PointNet* network architecture but sub-sample the point cloud by grid in advance and post-connect a CRF operator for fine segmentation.

The *PointNet++* is also the basis for many enhancements. *PointNet++* is based on a standard EDN, which additionally processes features of different abstraction levels (hierarchically). A notable characteristic of *PointNet++* is the use of the MLP as a central module. The enhancements of *PointNet++* aim on using the local neighborhood of the points as a feature. In order to describe the local orientation of the points, [239] add a *scale-invariant-feature-transformation* (SIFT) into the *PointNet++* as an intermediate layer. They refer this layer as *PointSIFT*, which describes the orientation of geometric features on different levels of abstraction. A *Local Spatial Aware* (LSA) layer has been developed by [240]. This can be used to model the feature distribution within an input set as a function. They construct the *LSANet* for this layer, implementing *PointNet++* for local geometric feature extraction. The *ShufflePointNet* is developed by [241]. Once again, a new layer is implemented into an existing *PointNet++*. This layer consists of a kNN based grouping of the input point set. For each subgroup, independent features are computed using an MLP. The new features are shuffled, concatenated, and fed into the next layer. Some of the MLP layers are replaced by this new layer in *PointNet++*.

In addition to developments directly related to *PointNet*, other point-based methods have been developed and a selection of some networks is briefly described below. The division of Figure 19, which subdivides RNN-based, point-based CNN and graph-based methods, is followed.

RNNs are often used in conjunction with the MLP network of *PointNet*, as shown previously. Alternatively, *RSNet* is a network where the entire point cloud is split into multiple views in x , y , and z directions. Each slice direction is processed independently. Slices are used to create an order in the point cloud. The features of the different slices are given into a RNN layer, sharing some of the information. The extracted features of the slices are aggregated and then the features of all slice directions are used for the point-wise semantic segmentation [242].

Conv layers convolve an ordered geometric neighborhood so that less data objects, such as pixels, carry more and more deep features. A transfer of this approach to individual independent points is targeted for point-based CNNs. Unlike the MLP, a *real* convolution is performed over a regional feature distribution. In *PointCNN* [243] this idea is described in detail and a χ -Conv layer is introduced. *PointCNN* uses an EDN. The coordinates are transformed into the feature space within a small region. This can be done in simple terms by computing the deviations of each point to a principal point. The principal point carries the area features and is combined with other principal points in the next stage.

A similar approach is applied to the *KPCConv* layer by [49]. As an initial step, the point cloud is homogenized using a grid-sampling filter. The point spacing is equal and the objects

become distinct by the features of the points. For example, points that are not occupied are marked with *free* or *0* and points that are part of an object are marked with the features of the measurement and *occupied* or *1*. The *KPCConv* layer in the basic version is built on spherical neighborhoods in which the correlation coefficients are calculated for all features and all points to the center of the sphere. The correlation coefficients can be multiplied by any weight matrix of the Conv layer such that they can be chained as EDN. *ShellNet* [244] also uses spherical input regions in which the features are computed. Circles with different radii are used and the features are processed together. *PointConv* [245] converts the local neighborhood of points into a continuous density function and a weight function that can be processed with Conv layer. The networks *A-CNN* [246] and *Dilated Point Convolutions* [247] are developed on point-based CNN in which the selection of points are optimized. *A-CNN* arranges the point cloud by a local projection of the sub-point cloud onto a disk and processes the features of the points with an encoding MLP network [246]. [247] perform point-wise classification with a CNN added by one FC layer and consider different receptive fields in the encryption phase. As an alternative to feature differences, which represent the relationship between points in a point cloud, graphs are widely used.

Graphs represent the relationship between data objects (e.g., points) through edges. Edges describe on one hand which data is in a relationship and on the other hand by the edge weights how this relationship looks. Thus edge weights are usually multidimensional vectors. Therefore, graphs take on an ordering role for a wide variety of data (e.g., social networks and point clouds) that are processed and analyzed with ANNs. A general overview of *Graph Neural Networks* (GNNs) is given in [248], which provides a clustering of the different types of GNNs. Simple graphs with few nodes are used for the joint processing of RGB- and D-images in order to use the local depth information, such as in the case of adjacent objects with similar RGB values [233]. GNNs in combination with MLPs have the purpose that point features and local neighborhood features are used together for depth feature extraction. In the *Feature-based Graph Convolutional Network*, initial point features are extracted and combined by the graph representation. Subsequently, a sub-graph convolutional network is build, which extracts neighborhood-based features. These depth-features are used for a point-wise classification by a FC layer [249]. A parallel feature decoding with graphs and individual points are described in [250]. For this, the individual features between the branches are shared at different hierarchical levels. The *Dynamic Capsule Graph* network architecture is based on an EN that processes at the input layer independently different feature types, such as *eigenvalues*, spectral values and coordinates. The features are combined and shared depth features are created by a chain of encapsulated graph Conv layers, which are summarized by pooling layers. The termination layers are MLPs, where a set of described features is generated for each point [51]. Other GNNs introduce an initial weighting of edge weights or features, during the encoding phase, to reinforce for the differentiation power of the relevant features [155].

The *EdgeConv* layer, representing the local neighborhood, is often implemented in MLP networks to minimize the disadvantages of *PointNet*. This *EdgeConv* layer is used for building

modeling [251] and in the analysis of ALS point clouds [252]. However, graphs can also be used for pre-segmentation of the full point cloud. In this case, the graphs are used to build sub-segments of those points that have a large feature similarity. These sub-segments can be processed separately in sub-networks [253]. Additionally, the results of these sub-networks can be used for context-based segmentation [254].

3 Connections of research publications

This section explains the connections between the four key publications. The connection with the ROs, the applied approaches and the general results are summarized in sections 3.1 to 3.4. The connections of the ROs and the publications are outlined in section 3.5.

3.1 PAPER 0: PCCT: A point cloud classification tool to create 3D training data to adjust and develop 3D ConvNet

The key topic of PAPER 0 is the development of a multi-user, browser-based tool for semantic segmentations of 3D point clouds. This tool is named PCCT¹ and consists of three independent modules. The data is exchanged via a *MariaDB* database hosted on a web-server. All three modules can be accessed from any computer within the *HafenCity University* (HCU) campus network without any local installations. The first module uploads new point clouds into the database. During the upload, the point clouds are converted into a 2D image representation. This 2D image representation is used for an automatic semantic segmentation and visualization within the browser-based tool. After the conversion, individual noise pixels are eliminated by filter algorithms and an edge optimization is performed. RG methods are used to create segments based on features, such as RGB and intensity values. Each image shows only one segment. The image is linked to the 3D cartesian coordinates of the corresponding points via a connection in the database. In the second module, these *segment images* are randomly loaded by the browser tool and the annotators use a drop-down list to select the appropriate class for the displayed segment. Thus, a class is assigned to each image respectively segment. The third module establishes the relationship between the cartesian coordinates and the classified images. Since each image is classified multiple times and by different annotators, a voting procedure is introduced to assign the most likely classes to the points. The point features are enhanced by the semantic classes. The enhanced point clouds are exported in different *ASCII* formats. Different projection methods and sets of features for segmentation are tested and applied in studies of PAPER 1 to PAPER 3. In addition, individual processing strategies are used for indoor and outdoor datasets, as they have different characteristics. A brief study is carried out to investigate the semantic accuracy of the PCCT. It is shown that the developed tool is suitable for the task, but an optimization of the tool parameters is necessary in order to achieve higher semantic accuracies. The optimization has to be done on the basis of characteristics which are defined in PAPER 2. The usefulness of manual semantically segmented point clouds is demonstrated on the example of an application with *PointNet*.

¹Github repository: <https://github.com/eb17/PCCT>

3.2 PAPER 1: Classification of erroneously measured points in 3D point clouds with ConvNet

The focus of PAPER 1 is to determine the influence of erroneous measured points in semantic segmentation with *PointNet*. Two datasets of TLS point clouds, which are created with the laser scanners *Imager 5010* and *Faro Focus 360*, are investigated in several experiments. The point clouds show outdoor scenes with ranges up to 100 m. Point clouds have varying point densities and a diversity of errors. Thus, these point clouds can be considered as worst-case data for semantic segmentation with CNN-based methods.

Methods to minimize unfavorable influences of the point cloud are discussed and some approaches and ideas on dealing with point clouds as input for CNNs are developed. The operation of DL methods, the requirements to start developing an DL application and the HP selection for semantic segmentation on the example of *PointNet* are described. Based on the research a workflow for semantic segmentation is developed and implemented by a *python* program using the *tensorflow Application Programming Interface (API)* [88, 255, 256].

The developments are examined in a study. The key objective of this study is to investigate a dataset as an information carrier. In different levels dataset-based, recording-system-based and point-feature-based information are investigated. The worst-case outdoor TLS point clouds that are manually annotated by using four different sets of class definitions are used for a study. The class definitions separate the point cloud into:

- class *Objects* and class *Erroneous points*,
- different object classes,
- all object classes and the class *Erroneous points*,
- the three largest object classes and the class *Erroneous points*.

This study shows that the erroneous points and inconsistent class sizes have an impact on the semantic segmentation result. The idea of a hierarchic class definition is concluded from this study. Additionally, the influence of the geometric extent of the segments is identified and discussed. These observations are the base for PAPER 3, which examines the dataset information in more detail.

3.3 PAPER 2: Evaluating the quality of semantic segmented 3D point clouds

The evaluation of the manual and the automatic semantic segmentation processes, as well as the semantic point cloud itself, are the key topics of PAPER 2. In order to develop an evaluation process, an assessment of available 3D semantic point clouds and annotation tools

is performed. The process from the recording to the semantic segmentation is investigated step by step. Whereby, efficiency, geometric and semantic accuracies are researched. The research findings are summarized in one set of the meta data of datasets. Unfortunately, the meta data for most datasets is not completely determinable. In some cases, the meta data can be determined from the data itself in an elaborate manner, but a lot of meta data remain unknown for third-party datasets.

In order to define and quantify which meta data about point cloud datasets is required, a quality model² is developed. This quality model describes the quality characteristics of semantic point clouds. The quality characteristics are

- availability,
- process reliability,
- completeness,
- consistency,
- correctness,
- precision,
- and semantic accuracy.

The description of each quality characteristic is realized by several qualitative or quantitative quality parameters. These quality parameters are binary and multidimensional statements / values about the point clouds, ratios, distances and standard deviations. The quality model is applied to the publicly available dataset collections and an evaluation matrix is used to present the quality of these datasets in a comparable manner.

The function of the complete quality model is demonstrated in this work by an self-created dataset, where all quality characteristics are described. This dataset contains indoor TLS point clouds which are recorded by the *Imager 5016*. As reference for the geometric characteristics, a high-quality point cloud is used. This point cloud is created with the handheld laser scanner *Leica T-Scan 5* in combination with the *Leica Absolute Tracker AT960* [257]. In the study two annotation tools, *Recap* and PCCT, are compared. It is discovered that the PCCT has an efficient and reliable process. The geometric accuracy of the PCCT point cloud is less accurate than point cloud processed by *Recap*.

The findings of PAPER 2 emphasize the importance of point cloud datasets for a reliable, efficient, effective and accurate semantic segmentation. The quality model makes the process and point cloud quality measurable. A transfer of the quality model on automatic semantic segmentations are co-developed, in such a way that the quality model is applied in the evaluation of PAPER 3.

²Github repository: <https://github.com/eb17/Quality-check-of-point-cloud-data-sets>

3.4 PAPER 3: Evaluation of class distribution and class combinations on semantic segmentation of 3D point clouds with *PointNet*

Semantic segments have different sizes and thus consist of different numbers of points. A semantic segmentation with DL methods works best if the information carriers (points) of the individual classes are equally distributed. This has been observed in studies with images [258, 259] and for 3D point clouds in PAPER 1. In order to quantify this observation and optimize the classification process of unequally distributed point clouds, investigations of class distribution and class size are performed in PAPER 3. Seven sets of *hard* and *easy* to separate class definitions are examined. In addition, oversampling methods, weightings of the input blocks, and weightings of the classification operation are analyzed, described, and further developed.

The workflow from PAPER 1 is enhanced by representing the input of the points as the local neighborhood in different point cloud blocks. The HPs of the *PointNet* algorithm are optimized with a dataset consisting of 76 million points and showing 27 rooms of the *HCU main building*. The influencing data-based HPs are identified and explained. The developed software tools³ for automatic semantic segmentation are divided into several modules in order to test different adaptations for optimization of non-uniform class distributions. Also a chaining of these methods is possible and theoretically different point-based DL methods, such as *PointNet++* [233] or *RandLA-Net* [44], can be applied.

Special attention in the definition of the classes is paid to the class *Erroneous points*, as it is identified in PAPER 1 as a possible influence to the semantic segmentation. The influence of this class could be confirmed. However, the presence of the class *Erroneous points* in a class definition can be beneficial or disadvantageous, as shown in the study. Additionally, the applied methods to increase the class equality could also increase the *recall* in most cases. The first objective for these study is that all classes are equally accurate. The second objective is, that *precision* and *recall* have to be better than 50%. The results of the study show that in most cases the accuracy has become more equal, but is still below 50%. These influences can be confirmed, but they are not the only significant ones.

3.5 Connections between the publications

The publication with peer-reviewed extended abstract (section 3.1) and three peer-reviewed (section 3.2 to 3.4) publications outline approaches, results, evaluations of research findings, and conclusions. The connections of the publications is illustrated in Figure 27.

³Github repository: <https://github.com/eb17/mypointnetworkflow>

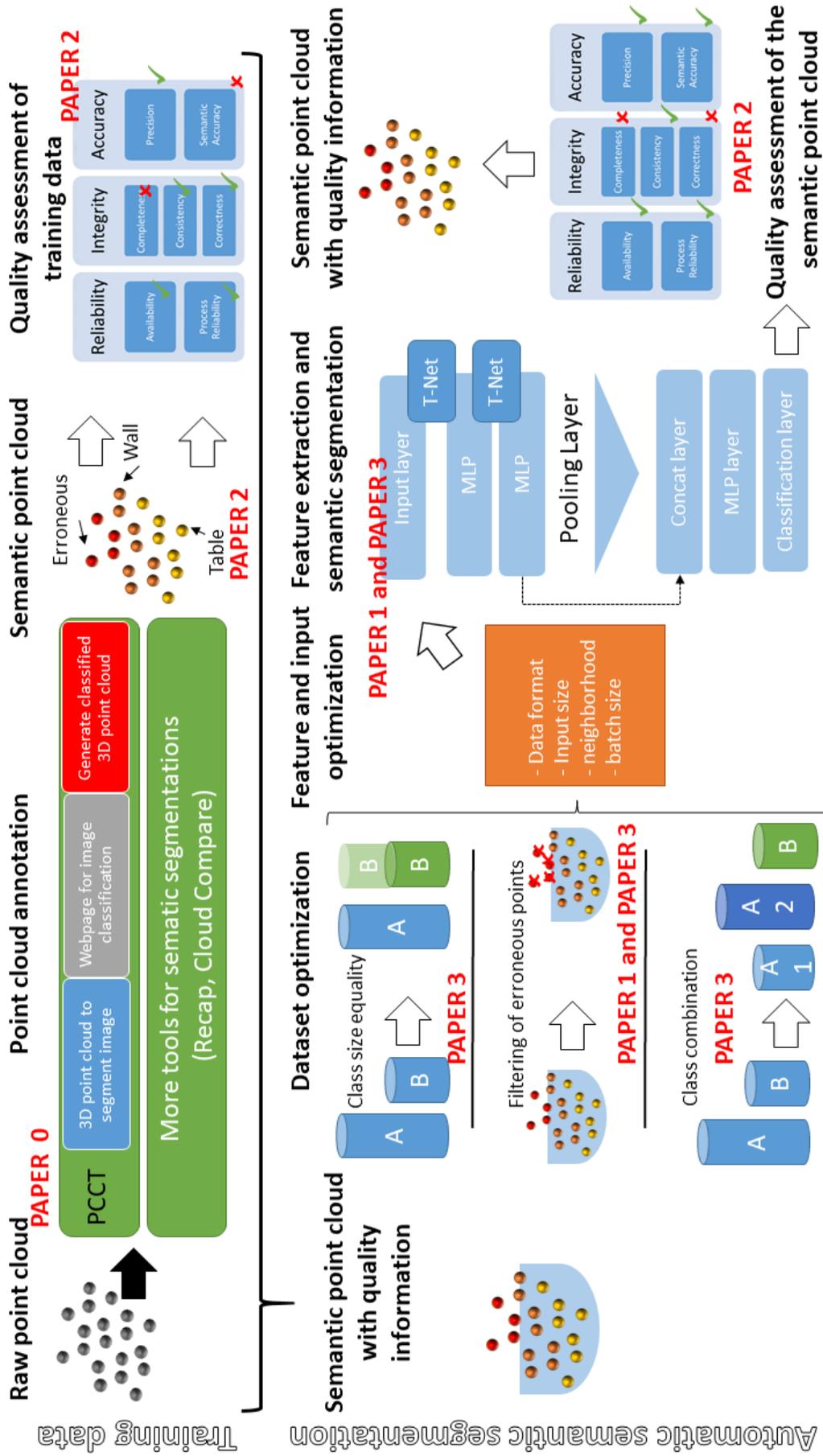


Figure 27: Overview of the connections of the four research publications. PAPER 0: Development of a browser-based classification tool. PAPER 1: Development of a workflow for semantic segmentation with *PointNet* and investigations on the influence of class Erroneous points. PAPER 2: Development of a quality model for the creation and evaluation of semantic point clouds. This quality model is used for the evaluation of the tool in PAPER 0 and the workflow in PAPER 3. PAPER 3: Extension of the workflow from PAPER 1 and development of methods to optimize the dataset for DL applications. Investigations of the development on the *PointNet* algorithm.

The PAPER 0 covers the development of a self-developed tool for efficiently annotating point clouds. The tool converts 3D TLS point clouds into 2D BEV, multi-view, and spherical representations which are automatically segmented. Using a browser-based tool, multiple users can simultaneously annotate the segments. In addition to the self-developed PCCT commercial and open-source applications for the annotation of point clouds are investigated. In the study of PAPER 2, the annotation processes and the annotation results are evaluated using the developed quality model. Applying the quality model on the semantic point cloud, additional quality meta data is added to each dataset. Based on the meta data of the dataset, it becomes assailable which performance can be achieved through the training of an automatic semantic segmentation algorithm with a particular dataset. The characteristics of each point cloud dataset describe the class distribution, the number of classes, the class definition and the degree of erroneous points. Since these characteristics are usually not ideal for DL algorithms, there is potential for optimization. In PAPER 1 the influence of erroneous points on *PointNet* is investigated. In the PAPER 3, further data-based influence factors, class definition, class equality, and class structure are examined. In the study of Paper 3, existing procedures to solve these issues are integrated into the workflow of the whole work and are evaluated. The development of a *PointNet*-based workflow for semantic segmentation of building recordings is described in PAPER 1 and PAPER 3. PAPER 1 represents the prototype of the workflow. The prototype is continually enhanced during the works progress. In particular the data pre-processing and the choice of HPs is improved. Supplementary details to PAPER 3 on graphs as input and hand-caved feature selections are outlined in section 5.2. In the study of PAPER 3 the automatically semantic-segmented point cloud is evaluated based on the quality model.

4 Evaluation of the research results

The results for three ROs are described in the section 4.1 to 4.3 by the nine RQ from section 1.3. For each research question, the methodological approach, the elementary development, the innovative findings, the conclusions, the outlook for further research as well as applications are described.

4.1 Development and evaluation of a tool for manual point cloud annotation

The first RO is the improvement and understanding of tools for creating semantic point clouds as training data. The focus is on the evaluation of available datasets, tools, processes (sections 4.1.1 and 4.1.2), and functions of the tools to enhance the development of annotation tools (section 4.1.3). In the case of manual semantic segmentations, the influence of humans is crucial, so it is investigated how humans can interact with different tools (section 4.1.4).

4.1.1 Survey of annotation tools for point clouds

RQ 1.1: Which annotation tools (manual segmentation) for point clouds exist? What functions can be found in these tools? How efficient, reliable and effective are these tools and how can these characteristics be determined?

Methodology: An intensive literature review of point cloud datasets and annotation tools is carried out using review papers and the key research papers [72, 162, 163, 169, 200]. A market analysis of services for classification tasks and software products in the construction, planning, and 3D visualization industries is conducted. Additionally, workflows and tutorials for open-source software are researched.

Findings: Point cloud annotation methods and processes are significantly influenced by the eight influences as explained in section 2.3 and shown in Figure 14. The recording system, the acquisition environment, and the application are the influences which are currently most considered in the creation process of training data point clouds.

Recording systems can record the scenes from a stable recording point or a moving platform (e.g., MSS), have different range limits and varying measurement accuracies. The scenes consist of various objects whose surfaces can be recorded accurately, with errors, or even not at all. In addition, obscuring objects prevent full coverage. The nature of the environment is also closely related to the application. A possible separation of application fields is usually made between indoor and outdoor applications. Outdoor applications mainly occur in the field of autonomous driving and indoor applications are most commonly used for reconstruction of buildings. This separation by environment is not strictly consistent, since facades are also often reconstructed from point clouds.

However, the application additionally specifies which semantic objects of the point cloud are determined and how they are geometrically selected. Typically, BB or irregular 3D solids are used in most datasets and tools. A summary of the annotation tools mostly applied to BIM and DL applications are presented in Tables 1 and 2 (section 2.3), as well as in Table 5 of PAPER 2. These tools are application driven with exception of *Cloud Compare* and *Recap*. Tools have been developed for one specific problem in connection with a semantic segmentation. Furthermore, these tools are mostly linked to one specific dataset. Datasets without a relation to a specific tool are rare. Some of these datasets are *Semantic3d.net* [40] and *TUM-MLS-2016* [153], which were created by *Cloud Compare*.

All tools divide the annotation into two steps. These steps are segmentation and labeling. For the most tools, labeling is a simple assignment of a class or an encoded class value to a previously created segment. This step is usually not automated. The more complicated step is segmentation. In the manual segmentation according to semantic aspects, the exact area must be shown in which an object can be separated unambiguously from the environment. This is particularly challenging if very large and very small objects occur. In addition, the object space must not be too large, so that a fluent navigation and visualization through a detailed point cloud is possible. In order to make this possible, the point cloud is usually divided into smaller sections in advanced, based on recording locations, rooms or distance intervals. Due to the complexity of the segmentation according to semantic requirements, errors often occur, so this step is topic for automation. The different methods for automatic segmentation and visualization are explained in section 2.5, as well as in PAPER 0 and PAPER 2.

The automation of the segmentation does not necessarily lead to more accurate semantic segments, but only to the fact that these are always determined the same. The determination and selection of the HPs for these segmentation methods is the central adjustment screw in the method development. The selection of the HPs is a very time-consuming task and must be carried out and checked for each individual dataset. DL approaches that only use geometric features are not published, even if e.g. *eigenvalues* and geometric parameters are suitable for this purpose, as shown by [184] for ML applications. More common semantic segmentation approaches are based on features, such as color and intensity values, since most datasets are created by RGB-D cameras and simple LIDAR scanners (Tables 2 and 3 of PAPER 2). These methods are also described in more detail in section 2.5 and PAPER 0.

The accuracy, reliability and efficiency are characteristics that can be used to compare different methods regarding a certain application. These characteristics can only be determined with effort and always for a specific dataset. Usually, third-party tools do not report these characteristics. In order to determine the accuracy, a reference point cloud showing the same scene and with a higher accuracy must be available. The simplest way to create such a dataset is to use synthetic point clouds derived from models [35, 166, 260]. Alternatively, a dataset can be created with a higher accurate and handheld measuring system as described in PAPER 2. In the considered case, a handheld scanner is used to create object-by-object segments. The combination of a very accurate recording system and semantic segmen-

tation in the field results in a reliable and accurate semantic point cloud. The creation is labor-intensive in the field and can usually only be applied to small point clouds, due to the usage of special measuring systems. If such a point cloud is available, a geometry comparison can be done to determine incorrectly segmented points.

In order to determine the efficiency of annotation tools, the required time must be put in relation to the achievable accuracy of the semantic segmentation. The costs for hardware, software and energy are mostly negligible, since the work of humans labor time causes the highest costs. Very few datasets or tools [41, 53, 54] indicate how long the semantic segmentation takes. Unfortunately, this information is given usually only for one annotator or a small group of annotators as discussed in RQ 1.4 (section 4.1.4). Caused by this missing information the efficiency can not be determined for most datasets.

Reliability is determined by multiple independent annotations and comparison with reference data. This does not require a GT dataset, but for comparability of tools, the same dataset should always be used. In addition, reliability in this definition also includes usability and describes how a certain group of humans solves the semantic segmentation task. Thus, aspects such as task comprehension and motivation can be included in this characteristic. For more details to this aspect see RQ 1.4 in section 4.1.4.

Conclusion and outlook: The available tools are highly specialized and not suitable for multi-disciplinary applications. Many innovative technical solutions are presented in the literature, but these require different input formats and lead to different semantic representations. Two basic functions, segmentation and classification, are available in most tools. These functions affect the quality of the semantic point cloud, these are discussed in RQs 1.2, 1.3 and 1.4. Meta data is incompletely obtainable for many datasets and annotation tools, and a comparison of them is often not possible. The evaluation of point cloud datasets and annotation tools is addressed in RO 2. This analysis is very time-consuming, but necessary for a better understanding of point cloud data and algorithms, as seen from the first investigations to determine the three most discussed characteristics.

4.1.2 Annotation tools for indoor terrestrial laser scanning point clouds

RQ 1.2: Which annotation tools can be used for the semantic segmentation of challenging real-world indoor TLS point clouds?

Methodology: Annotation tools and processes were selected based on the literature review explained in the answer of RQ 1.1 in section 4.1.1. The tools *SemanticKITTI* [53], *PC-Annotate* [55], *Recap* [134], and *Cloud Compare* processes from *TUM-MLS-2016* [153] and *Semantic3d.net* [40] are tested with the reference dataset published in PAPER 2. In addition, the PCCT developed in PAPER 0 is evaluated to investigate its quality. In a pilot study, all tools are evaluated by two volunteers in terms of usability (data format, availability of the software, technical requirements, approximate processing time). In the main study, ten volunteers are asked to perform semantic segmentations with *Recap* and PCCT (pub-

lished in PAPER 2). In a survey, prior knowledge, metrics (e.g., processing time), impression of usability, expected accuracy, and desired changes are asked. The survey is evaluated together with the results of the semantic segmentation.

Findings: The results of the pilot study shows that *SemanticKITTI* and *PC-Annotate* are not suitable for semantic segmentation of challenging real-world indoor TLS point clouds. The annotation tool of *SemanticKITTI* is optimized for mobile recorded input data and for dynamic-changing environments. For a semantic segmentation of TLS point clouds, a pseudo navigation file would be needed in order to use this tool. The *PC-Annotate* tool has a limited selection of classes and a semantic segmentation is only efficiently possible via the fit of regular geometries, which leads to inaccurate segmentations for detailed indoor scenes. The processes of [40, 153] are complex and require a solid knowledge of *Cloud Compare*, which cannot be assumed for all potential annotators. Therefore, these tools were excluded from the main study.

The main study is presented in section 4 of PAPER 2. All quality parameters of the model from section 4.2 and in section 3 of PAPER 2 are determined, so that among other parameters the accuracy and the efficiency are evaluated. The study results demonstrate that the most accurate semantic segmentation is preformed by annotation tools with a free choice of the perspective and a lasso function for segmentation. The annotators work very detail-oriented, which leads to an extended processing time and a decrease in efficiency. The effectiveness is higher with a tools such as *Recap*. In general, the PCCT, which only allows the annotator to classify, is very efficient, but for very small objects it is not effective. Measurement errors in point clouds make the segmentation difficult for any tool, because boundaries between an object and points representing mix-pixel errors, defuse reflection and comet tails cannot be clearly identified. Geometry-based filtering can make manual and automatic semantic segmentation effective, reliable, and accurate. The segmentation of a coarse pre-segmentation of rooms allows *smoother* navigation through the point clouds. Current hardware reach its limits for processing very large point clouds with such tools, because the working memories are not large enough.

Conclusion and outlook: The research of the available annotation tools shows that only few tools are suitable for the application of modeling indoor rooms. To the best of the author's knowledge, a systematic evaluation of these tools has been carried out in PAPER 2 for the first time. An annotation tool that can be used across various disciplines is urgently needed. A basis for this development can be the PCCT. Point cloud annotation tools from commercial service providers and CWS are not investigated in detail due to the lack of transparency regarding costs, data security, data rights, and working conditions of crowd-workers. After all, the commercial tools are the drivers for many applications in which semantic point clouds are needed.

4.1.3 Development of an annotation tool

RQ 1.3: How can semantic segmentation tools for point clouds be enhanced and improved?

Methodology: At the beginning of this research¹, few scientific tools for semantic point cloud segmentation were available. Commercial service providers predominantly offered semantic segmentation for image data. Commercial and open-source offline software for point cloud processing, such as *Geomagic Wrap* [261], *Cloud Compare*, and extensions for CAD programs, are the state of the art. An adaptation of existing systems to improve them is not technically purposeful, therefore the complete development of the PCCT is necessary. A concept for data management, implementation of segmentation methods, classification and visualization is developed on the basis of the analyzed annotation tools from RQ 1.1 in section 4.1.1. The PCCT is iteratively developed and evaluated in studies that are explained in the RQs 1.4, 2.1, 2.2 and 2.3.

Findings: The annotation tools available on the market show potential for further developments regarding issues such as data security, capability of a dataset for multiple-users, segmentation and classification functions, and automation of time-consuming sub-operation steps (Figure 28). These issues are considered during the development of the PCCT. The PCCT is an experimental tool, which can be used to optimize the issues and evaluate the experiment.

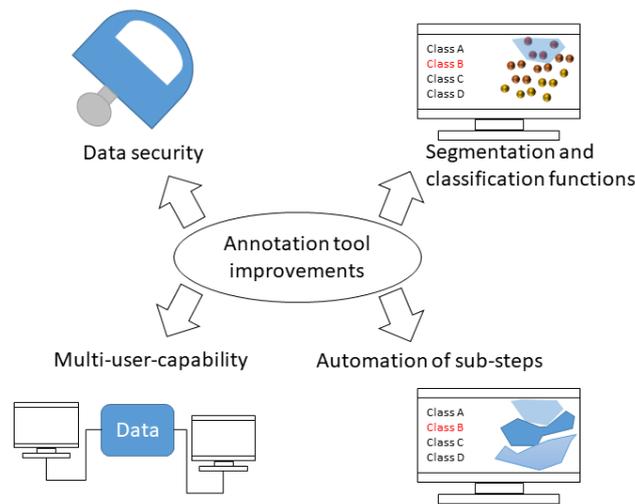


Figure 28: Central issues for improvement in available point cloud annotation tools: Data security, multi-user-capability, segmentation and classification functions, and automation of sub-operation steps.

Point clouds are detailed representations of real buildings and make hidden information visible. This information must be kept safe to third-party access for critical infrastructures, such as for port facilities, utility lines, airports, prisons, railroad facilities, or research facilities.

¹In 2017

The usage of CWSs is usually not possible for these infrastructures [55]. Processing of a large dataset by only one annotator is also in most cases not possible and can also lead to classification bias in the semantic point cloud. Therefore, it is necessary to store the data in such a way that it can be accessed in parallel. In the best case, only parts of the point cloud are made available to the user, allowing to edit but preventing understanding the entire infrastructure. The classification bias can be minimized by having different annotators to perform the semantic segmentation. Based on these considerations, a database-based concept is developed for the PCCT, as published in PAPER 0. There, a copy of individual point cloud sections are provided for processing via a browser-based tool. The results of different annotators and classification passes are connected to the original point cloud, but a final assignment of the semantics to a point is made after max-voting over all classifications.

The class definition of the semantic segmentation must be transferable into the annotation tool. In order to use the annotation tool in almost any application the list of possible classes

- must be re-defined in each case.
- include very general classes, cover a large amount of classes.
- has a hierarchical organization.

A customizable list of classes in the tool can lead to the fact that the class definition is no longer unique. A very general list limits the application scope and a list with too many classes can no longer be overlooked, which can lead to different understandings by the individual annotators. For example, if the class *Wall* and *Facade* are available, it is not always clear how they differ. For the PCCT, it is experimented with a class definition that is as general as possible and specialized for building parts and furnitures. Even with this list, a confusion can be seen in the study results due different understanding of classes. The hierarchical organization of the classes using e.g. the *WordNet* scheme [262] is a technique, which allows a maximum of variety and uniqueness. The technical implementation and the usability of this variant is complex, since an effective navigation must be applied for a list of several thousand words. Currently, list of classes that can be created by an administrator are most effective for practical applications.

Some approaches of automation for segmentation of point clouds are described in section 2.3. The special characteristic of TLS point clouds is that they are recorded from a fixed point of view and usually represent a 360° view of the scene. A transformation of the cartesian coordinates into polar coordinates is directly possible. The fix angular increments of the polar coordinates allow a transformation of the 3D point cloud to a structured 2D image. The distance measurements can become a feature variable. Graph-based methods in 2D applications, such as [95, 263], have a high degree of development and provide unique segments. Moreover, 2D segments can usually be visualized and interpreted better by humans than 3D segments. Other developments use BEV approaches of [4, 27] for a semantic segmentation of buildings after the removal of ceilings and floors. These approaches use a floor by floor representation of the building. A disadvantage of the 2D projection is that the geometrical depth is usually not considered in the segmentation step and a distortion oc-

curs. Different radii and projections have been tested in the PCCT to perform segmentation automatically and accurately ([264] as well as in PAPER 0 and PAPER 2).

Conclusion and outlook: The development of an experimental prototype annotation tool for semantic segmentations is implemented with the PCCT. The PCCT is based on a transformation of the 3D points to 2D pixel, which is partially disadvantageous for some of the applications. Different influencing variables can be tested with the modular-structured PCCT. An annotation tool for any kind of applications and that fits to all requirements from above is not available on the market yet. The optimization of annotation tools is still important but an under-researched topic for more accurate and reliable semantic point clouds.

4.1.4 Human factor in semantic segmentations of point clouds

RQ 1.4: How to become a good annotator for semantic point clouds? How can the performance of annotators be measured? What do annotators need and how can the tool support them?

Methodology: The influence of the human annotator is a part of the study in PAPER 2 and is determined for the PCCT and *Recap*. Quantitative parameters such as processing time, precision, and accuracies are measured or calculated by means of the high-quality reference point cloud dataset. Qualitative characteristics are assessed by a questionnaire that is answered before, during, and after the task by the volunteer annotators. Additionally, it is asked for a self-assessment, previous experience and a descriptions of how the tools were used.

Findings: The fact that the annotator has a key role for the quality and usability of a semantic point cloud has been noted by [40, 55]. [54] developed valuable rules of thumb for the selection and training of annotators. Feedback during the annotation is given by feedback function in the tool of [265]. Humans are very good at recognizing varying shapes of objects [266]. Following these ideas, a study is conducted, whose results describe the human influence. Guidelines for processes and tool developments should result from this. The ten volunteers of the study had no, little, medium or very much experience in the handling of point clouds and the semantic enhancement of point clouds in advance. This previous knowledge allows an evaluation of annotators training, developed execution process and tool functions.

Training documents are prepared for each investigated tool and are given to the volunteers in advance. These documents are assessed and the volunteers have to paraphrase the task in their own words. By this first task, it could be determined that illustrations contribute to a better understanding of the task. The ideas of the application and of point clouds as well as its interpretation variate strongly between the annotators. The given feedback is used to improve the training documents with example images and detailed class descriptions. Helpful in the class description is to clearly include or exclude objects that are geometrically or semantically similar to others. As an example: A door consists of the frame, the leaf and the handle, but not of the window next to the frame. Different volunteers have examined the documents at different stages of the development before the finale experiment.

The training and the feedback process are planned in detail based on review of the literature. A few days before the experiments, all documents (task description, class definition and illustrated instructions) are given to the volunteers. In order to get familiarized with the task and to answer the first part of the questionnaire. Before the annotation experiment, the tools are explained and questions could be asked. The annotation is done without any supervision. All volunteers are able to solve the tasks. The average class accuracy for most annotators is above 90% (*recall* and *precision*). Large differences can be observed among the different classes, such as *Floor* and *Ceiling* are above 95% for the parameters *recall* and *precision*. The infrequent classes *Chair* and *Table* are less accurate and usually vary between 85% and 95% for *recall* and *precision*. The percentage of TP points in the class *Erroneous points* is usually lower than 50% (*precision*), because in case of doubt object points usually become erroneous points. Large variation is found in the time required. Some volunteers finish within less than 9% of the maximum time. Large differences in time are also found between the tools. The PCCT is more efficient than *Recap*, but the results are not as correct and the simplicity of use is perceived as tedious. The usage of *Recap* is complicated at the beginning for some volunteers, so errors occurred more frequently due to the segmentation functions and the processing takes a long time. Point density and variety in the activity are seen as particularly important, in addition to a clear task description. The navigation through the point clouds and self-dependent segmentation, such as using a lasso, are functions that make this possible.

A relation between high efficiency and correctness cannot be found. The volunteers find it helpful to be able to ask questions during the task, since there are occasional misunderstandings or ambiguities. These results are taken from the study in PAPER 2.

Conclusion and outlook: A good annotator does not need to be an expert in point clouds. Unique task descriptions, class definitions, and continual feedback are most critical for a successful semantic segmentation. Exclusive classification tools, such as PCCT, level the entry threshold, but leads to tiring with very large datasets.

4.2 Development of a quality model for heuristically describing semantic point clouds

The findings to the second RO makes the quality of a semantic point cloud measurable. For this purpose, the characteristics of the point cloud are investigated (section 4.2.1) and a quality model is developed (section 4.2.2). In order to use the quality model, it is transformed into an evaluation matrix with which semantic point cloud datasets, annotation tools and automatic workflows can be evaluated (section 4.2.3).

4.2.1 Point cloud quality

RQ 2.1: What are suitable semantic point clouds? What are the characteristics of point clouds? How can the characteristics of the point cloud be determined, measured and compared?

Methodology: A literature research of the characteristics of point clouds has been performed and the creation process of a manual and automatic semantic segmented point clouds has been analyzed on the *HCU main building* dataset.

Findings: The definition for a suitable semantic point cloud, which is developed as a result of this work, is given by the following statement.

Definition 1: *A good semantic point cloud has a homogeneous density, is free from data gaps, measurement and registration errors, the geometry of the semantic segments corresponds to the objects in reality and the labels accurately describe the object semantics.*

A semantic point cloud that completely fulfills definition 1 usually does not exist, so that the degree of individual characteristics are determined. This is necessary because point clouds with low quality levels are not suitable for some applications as discussed in RQ 2.2 (section 4.2.2). Before the quality can be determined, the creation process and the characteristics of the semantic point cloud, as well as a definition of errors must be stated:

Definition 2: *Errors are the influences that lead to not fulfilling the definition for a suitable semantic point cloud.*

The creation process of semantic point clouds usually consist of the recording, the registration and a subsequent segmentation according to semantic properties of represented objects. The geometric correctness with respect to the recorded surface is the most studied characteristic for the recording and registration [81, 82, 83]. The parameters *standard deviation* and *deviation from a target geometry* are typically used to describe the characteristic geometric correctness. Semantic segmentation is the creation of an abstract model from the data model (measured values) and the knowledge about the real world. Such a process is defined in [267] and shown in Figure 29. As additional information the knowledge of a human annotator or the DL algorithms is used to semantically enhance the point cloud. The accuracy of the semantic segmentation is described by the performance of the selected manual or automatic algorithm. This is usually expressed in terms of a ratio of incorrect and correct data objects (pixels and points). The parameters *precision*, *recall* or *IoU* are commonly used. The geometric shape is not expressed by these parameters.

Varying error descriptions are found in publications about semantic point clouds. Measurement errors occur in the form of interference points and noise around a surface, which are analyzed for the recording system. In the process of semantic segmentation, these points become an additional semantic class. They are no longer errors for the semantic segmentation. Errors in semantic segmentation are points that are assigned to the wrong class. The definition of errors change in the two-step process. However, it is necessary to consider the

errors from the previous step in the following one. In the final point cloud, the geometry and semantics of the point cloud should be correctly represented. The recording and registration accuracy are usually not used in semantic segmentation. They are relevant for deriving geometries and models from the semantic point cloud, so that the objects are not distorted and show the geometry.

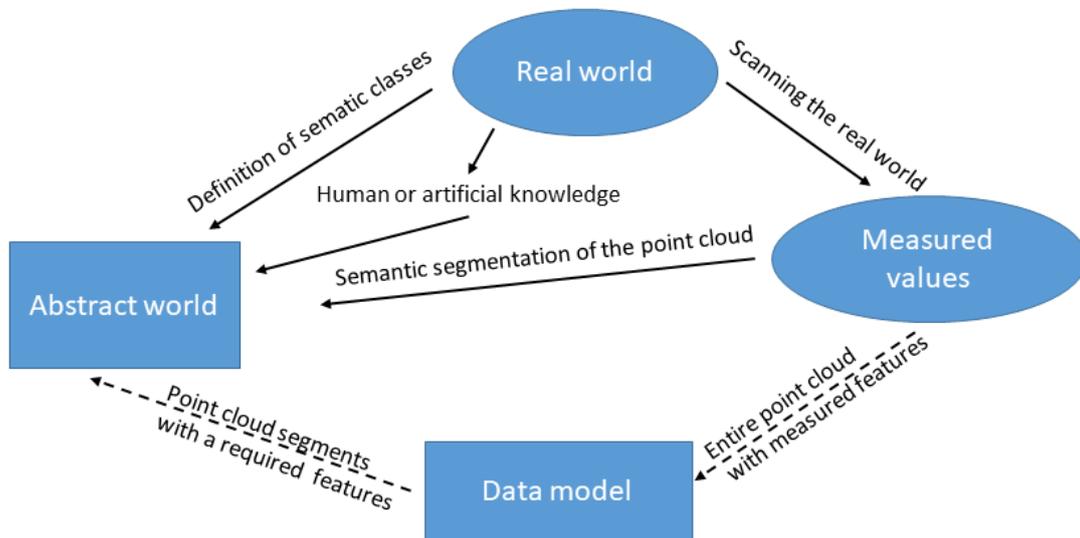


Figure 29: Process of semantic segmentation of point clouds serving as an abstract model of the reality. Taken from PAPER 2 and adapted.

For an evaluation and usage of the point clouds, not only just discussed characteristics geometric and semantic precision, as well as correctness are relevant. Other characteristics are:

- the availability of data and metadata,
- the process reliability,
- the completeness of data and processing,
- and the consistency of data content (e.g., type of variables),

as explained and developed in PAPER 2. In order to determine and compare the characteristics, the developed quality model is an effective tool. For each characteristic, qualitative and quantitative parameters are defined. The development of the set of quality parameters is described in more detail in PAPER 2 and is discussed in RQ 2.2 (section 4.2.2). Thus the comparison of different semantic point clouds is possible. In order to determine the degree of quality of the point cloud for an application, threshold values must be defined for each quality parameter.

Conclusion and outlook: The quality of a semantic point cloud is determined by different steps, which are usually concatenated. The definition of what is an error changes from step to step. In order to consider all errors in the final semantic point cloud, these or a description

of these must be passed on at each step. The quality of a semantic point cloud consists of many different characteristics, which are determined in the individual stages. Structuring them into a quality model developed specifically for semantic point clouds is effective and implemented.

4.2.2 Quality model for point clouds

RQ 2.2: How is a quality model for semantic point clouds designed? Which parameters are necessary for the description of the characteristics? Does the quality parameters differ for annotation and automatic semantic segmentation?

Methodology: The quality model of [92] is used as the basis of the quality model for semantic point clouds. The characteristics from RQ 2.1 (section 4.2.1) represent the structure of the quality model. In the course of the annotation process the descriptive quality parameters are determined and evaluated.

Findings: The developed quality model describes seven characteristics of the point cloud which are directly or indirectly related to the semantic segmentation. Direct related characteristics are accuracy and precision. For example, indirect characteristics are usage constraints, such as for the datasets of [53, 64, 226, 268] which are only allowed to be used as a benchmark.

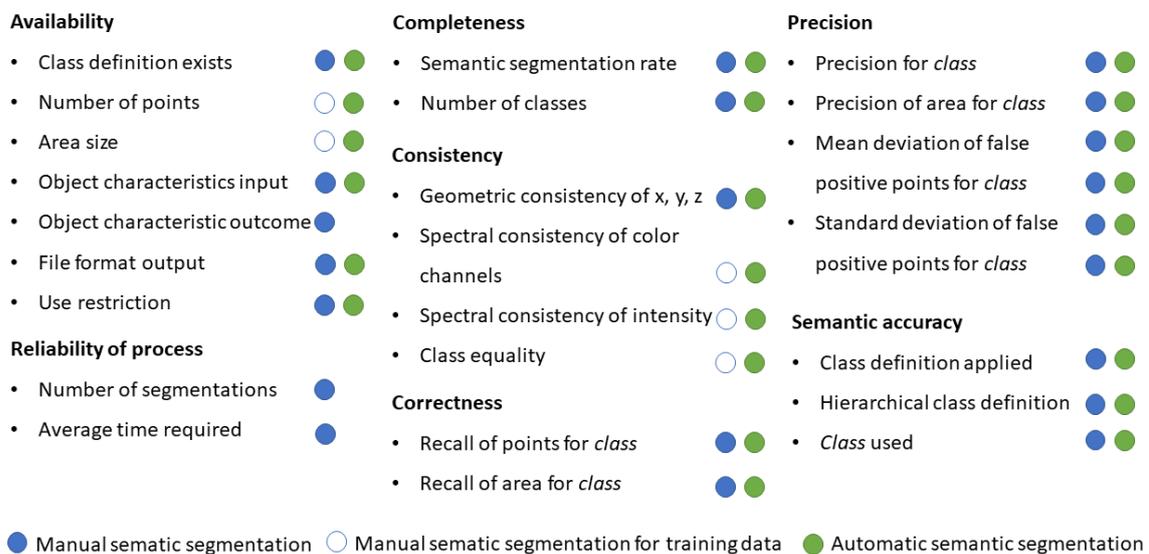


Figure 30: Quality model for semantic enhanced point clouds. Seven relevant characteristics with descriptive quality parameters are shown. Classification of necessary parameters for: Manual segmentations (filled blue circles), manual training data generation (unfilled blue circles) and automatic semantic segmentation (filled green circles).

The characteristics become measurable and comparable by quality parameters. The quality parameters of the developed model are summarized in Figure 30 with the respective characteristics. Some quality parameters are in a relation to others, so that e.g. the *precision*

of the area can only be determined if information about the area is available. Most quality parameters on the left side of Figure 30 are the independent parameters and on the right side are the dependent parameters. Not all parameters can be determined for every point cloud, as noted in section 4.2.1. However, it is possible to use the quality model with fewer characteristics if they are not relevant for the task or application.

The choice of quality parameters is influenced by applications in which models of real buildings and descriptions of urban space are created. Thus, not only the (semantic) point cloud itself is considered but also the purpose of the final product. Quality parameters, such as number of points, input variables or data formats, are necessary for any semantic segmentation. The quality parameters are defined and explained in detail in section 3 of PAPER 2.

Not all quality parameters are necessary for the manual or the automatic semantic segmentation. The number of points, the area size of the classes, the spectral channels and the class equality are not mandatory for manual the semantic segmentation if a model or an application is built from the point cloud. However, these characteristics are important if the point cloud is used as training data on an automatic semantic segmentation. For semantically enhanced point clouds that are not used as training data, the spectral outputs, the amount of segmentation runs (usually only one), and the duration of the process, are usually of little or no meaning. This leads to the fact that the process reliability characteristic is no longer necessary in the used quality definition.

Conclusion and Outlook: The quality model is the framework for the evaluation of semantic point clouds. The characteristics and its definition by parameters vary based on the application and used semantic segmentation method. The developed quality model is flexible enough to be applied in manual and automatic processes due to the quality parameters. Regarding process reliability in automatic processes, an enhancement of the parameter set is necessary. The main applications for the developed quality model is in the semantic segmentation of point clouds for buildings.

4.2.3 Application and evaluation of the quality model for building point clouds

RQ 2.3: How can the quality model be applied for the semantic segmentations of building point clouds?

Methodology: The explained quality model from RQ 2.1 and RQ 2.2 (sections 4.2.1 and 4.2.2) is applied to a selection of publicly available and private semantic point cloud datasets. All meta data is researched, as far as it can be determined. In addition, the quality model is tested in two point cloud annotation tools and on an automatic semantic segmentation workflow. Here, the purpose of the semantic segmentation is always to use the point cloud as a basis for building reconstruction.

Findings: The usage of the quality model is described in PAPER 2 and PAPER 3. The results of the evaluation of datasets, tools and a workflow are shown there. The central statements for applying the model are summarized in this section.

Conclusion and outlook: The developed quality model can be used in the form of an evaluation matrix for the evaluation of datasets and as development stages of annotation tools and automatic semantic segmentation workflows. A simple implementation can be realized with the implemented excel sheet. An implementation in the form of a web database could expand the user community and is desirable. An effective selection of thresholds remains a challenge, which is addressed in PAPER 2 by fundamental ideas.

4.3 Development of a workflow for semantic segmentation to investigate datasets and point features as influences

The choice of algorithm (section 4.3.1), its HPs (section 4.3.2) and data-based HPs (section 4.3.3) are investigated in this RO. For this purpose, a workflow is developed in which these parameters are studied efficiently and effectively.

4.3.1 Workflow development and algorithm selection

RQ 3.1: How can DL methods be integrated in a workflow for semantic segmentation of point clouds? Which DL methods are suitable?

Methodology: A literature review of methods for automatic semantic segmentation using ML and DL approaches is carried out. Based on the review, methods are selected and applied for experiments where point cloud scenes representing indoor environments and street sections of up to 400 m² are used. The workflow is developed based on these investigations and the review of common APIs.

Findings: The suitable methods are summarized in PAPER 1, PAPER 3, and in section 2.6. The semantic and geometric accuracy of the method and the feasibility for larger datasets are the key factors for a reliable 3D model. Based on the literature review on methods for automatic semantic segmentation using ML and DL approaches, point-based methods are most suitable for TLS scans of indoor environments and of small road sections, as the following constraints have to be considered:

- The density of the point cloud changes depending on the measuring system.
- Multiple point clouds of points of recording can be combined to one point cloud.
- Very detailed and at the same time very extended objects need to be segmented.
- Objects usually expand in all three directions.
- Semantic objects can have gaps in the representing point cloud caused by occlusions during the recording.
- Point clouds have a high percentage (of up to 10%) of erroneous points.
- Color values for the point clouds are not mandatory.
- In these application the intensity values mainly depend on the angle of incidence.

Point-based methods do not perform any generalization on the input format, use the geometric point cloud distribution in a certain section and usually do not rely on spectral variables. Erroneous measurements that are geometrically close to objects can thus be differentiated, and do not become objects as in 2D projection-based and 3D grid-based DL methods. Gaps in the data have a small impact because they are learned along with the data. More problematic is that most methods can only consider the local neighborhood or a very thin point cloud. Methods that combine information from different input neighborhoods into a DL algorithm are reviewed in section 2.6 and an alternative approach using an *adjacency matrix* is shown in section 5.2.

The workflow has the function to unify and combine measured point clouds, which are in different formats, with different variables and in different variable ranges. This data has to be transformed into an input being suitable for the algorithm. The transformed data format are usually graphs and sub-point lists organized in batches. The output of the semantic enhanced point cloud must be in a format in which it can be further processed. In the workflow, the algorithm is used in three different modes:

- *Training mode* in which point clouds and GT labels are used.
- *Evaluation mode* in which the algorithm can make a prediction based on the point cloud features. This prediction is evaluated by means of the GT-Label.
- *Application mode* in which the algorithm makes a prediction based on the point cloud features.

The basic elements of the workflow are shown in Figure 32. This concept can be applied to all point-based methods which are explained in RQ 3.2 (section 4.3.2) using *PointNet* as an example network architecture.

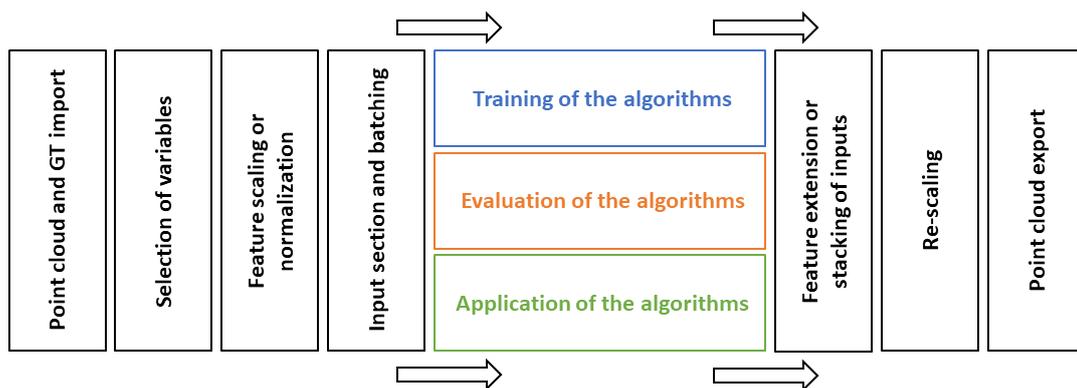


Figure 32: Concept for a workflow to apply DL methods for semantic point cloud segmentation. Three modes for the procedure: Training, evaluation and application.

Conclusion and outlook: Point-based DL methods are applied on TLS point clouds with semantic objects, whose shape is variable, are currently the most suitable semantic segmentation methods. The concept of data flow is applied in the training, evaluation and application mode. A workflow also makes it possible to compare different HP combinations with less effort.

4.3.2 Hyperparameter selection for point-based deep learning methods

RQ 3.2 Which hyperparameters need to be defined for applying *PointNet* in a semantic segmentation workflow? How are the values for these hyperparameters determined?

Methodology: The development and experiments are mainly carried out on the *PointNet*. As with any DL architecture, there are HPs that must be defined prior to application. The initial value ranges for the HPs are determined using rules of thumb, experience from previous experiments, and recommendations from the current literature. A *cross-validation* procedure is used to determine and apply the optimal values for these ranges. This procedure is still commonly seen in [130, 270].

Findings: *PointNet* is one of the first network architectures to directly semantically segment point clouds without format conversion. Nowadays, the performance of a pure *PointNet* architecture no longer corresponds to the state of the art accuracy. However, *PointNet* layers and its basic concept, the use of a MLP for independent deep feature extraction, are applied in many high-end DL methods. *PointNet* is thus a key network architecture and is examined and explained in PAPER 1 and PAPER 3. Each network architecture has general and specific HPs that are affected by the data, the software implementations, and the used hardware. The details of the applied hardware and software parameters are summarized in Table 3. The investigation hardware is a high performance workstation in the medium price range, since the developments should be economically transferred into practical surveying applications and projects. For a few and very complex investigations, a GPU rack with ten *Nvidia Tesla V100 32 GB* GPUs is used.

Table 3: Parameters of the hardware and software used for development and testing (single workstation).

Hyperparameter	Name / Type	Version	Time / Amount
CPU	AMD Ryzen Thread.	2970WX	1
GPU	GeForce	RTX 2080 Ti	1
ROM	SSD		475 GB
RAM			64 GB
RAM GPU			11 GB
DL-Framework	Tensorflow	2.3.0	
Prog. Language	Python	3.8	
GPU Accelerator	CUDA	10.1	
Training duration			> 1 to 72 hrs

The general HPs must be defined for each network architecture and thus define the characteristics of the architecture. These HPs include the *type*, *number* and *combination of layers*. Other general HPs include the LR, *type of weight initialization*, *dropout rates*, *the number of epochs*, *batch sizes*, *metrics*, *early-stopping criteria*, *optimizer*, *loss* and *actuation level functions*, the *classification function*, *transfer learning* (TL) methods, and the strategies for optimizing all these parameters. Specific HP involve the form and number of *data inputs*, as well as the *selection of feature variables*. For the *HCU main building* dataset, HPs were

determined as an example. The general HPs, as finally applied, and the ranges are shown in Table 4. The *PointNet*-specific HPs are presented in Table 5.

Table 4: General HPs for CNN architectures. Optimized set of HPs and typical values ranges for these HPs.

Hyperparameter	Used settings	Common settings
Layer	MLP, max pooling	MLP, FC and max pooling
Loss function	Categ.-cross-entropy	Categ.-cross-entropy with logits
Classifications function	Softmax	Softmax
Activation function	ReLu	ReLu, sigmoid
Dropout rate	0%	Up to 30%
Optimizer	Adam	Adam, momentum
No. epochs	50 or 100	25 to 500
Batch size	16	4 to 64
LR	0.001 to 0.00025	> 0.001
Adaption of LR	Yes, step-wise	Any 300,000 steps by 50%
Weight initialisation	random fix	Random, random fix, TL
Indep. trainings	9	Not published
Transfer Learning	Iterative	Yes, no or iterative
Metrics	(Eval.-)loss and accu.	(Eval.-)loss and accu.

Table 5: *PointNet*-specific HPs. Optimized set of HPs and typical values range for these HPs.

Hyperparameter	Used settings	Common settings
No. of input points	1024	1024 to 4096
Input size	$1 \times 1 \times 6$ m	$1 \times 1 \times \infty$ m
No. of feature variables	9	3 to 12

In order to determine the HPs, several training passes are carried out on a representative subset of the dataset (three rooms). In each training pass, one HP is changed incrementally. This procedure is afterwards repeated for the next HP. In order to accelerate this optimization process, prior experience and estimations are used. For example, the size of the input cubes can be estimated from the object sizes in the semantic classes. These HPs represent only parts of the adjustment parameters, but must always be taken into account in DL applications. Further *data-based* HPs (DHPs), are only investigated in very few Researches, as in [250, 258, 271, 272].

Conclusion and outlook: The typical ranges of HP values for a *PointNet* architecture are determined theoretically and empirically, for which test workflows are programmed. For a TLS indoor dataset, the *optimal* HPs could be determined. These HPs represents the basis configuration for the DHP in RQ 3.3 (section 4.3.3).

4.3.3 Data pre-processing and data influence

RQ 3.3: How can the influence of the dataset be controlled by data-based hyperparameters in the semantic segmentation of point clouds? What are the main influences?

Methodology: The dataset, whose characteristics are described and controlled by the DHPs, is an important influencing variable for semantic segmentation of point clouds. The characteristics, the class distribution, the class definition and the incorrectly measured points are analyzed and evaluated by empirical investigations with the developed workflow of PAPER 1 and PAPER 3.

Findings: The DHPs are set by the dataset. Examples are the available feature variables, the class distribution or the size of the dataset. A collection of the most common DHPs is summarized in Fig 33. An overview of the structure of the dataset, the semantic content and the features allow a systematization of the influences and its investigation. The semantic DHPs have been investigated in RO 2 (sections 4.2.2 and 4.2.3). The selection and local computation of geometric and spectral features are studied for general ML methods in [39, 273]. In addition, the dataset size, the normalization of features, and the density of the point clouds are considered in many network architecture developments [37, 200, 274]. Rules of thumb can be derived from this, but they are not supported by any systematic proof. The structure of point cloud datasets is usually only investigated with respect to the input formats. With few exceptions [172, 250, 271, 275, 276] class definitions, characteristics of erroneous points, and class size differences are not considered in point cloud datasets, even though these are considered to be a well-known influencing factor in semantic segmentation of images [258, 277]. These three DHPs are explored in PAPER 1 and PAPER 3.

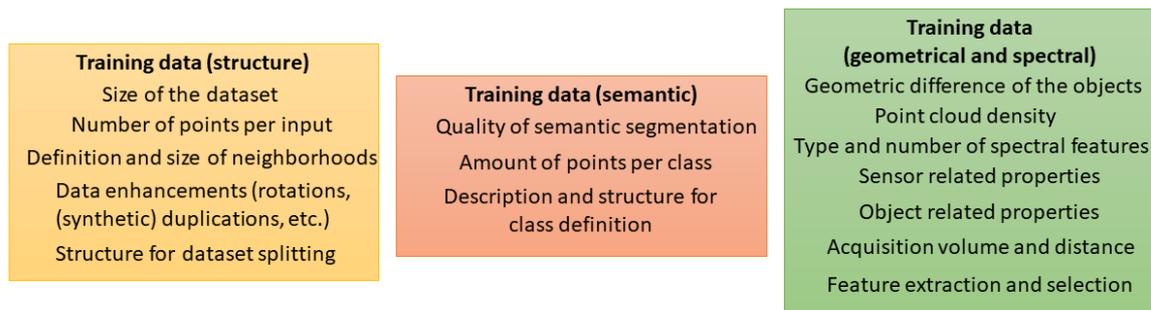


Figure 33: DHPs for semantic point clouds. The DHPs can be distinguished according to structural, semantic, geometric and spectral characteristics. A selection of the most common DHPs for each property is summarized.

The class *Erroneous points* is usually determined less precisely by most algorithms for semantic point cloud segmentations than the object classes. This is shown by the analysis of point-based CNN at the leader board of the TLS dataset of *Semantic3D.net* [40] (Figure 34). In addition, this analysis shows that more frequent classes, such as *Building* and *Road*, are determined more accurately than the smaller class *Tree*.

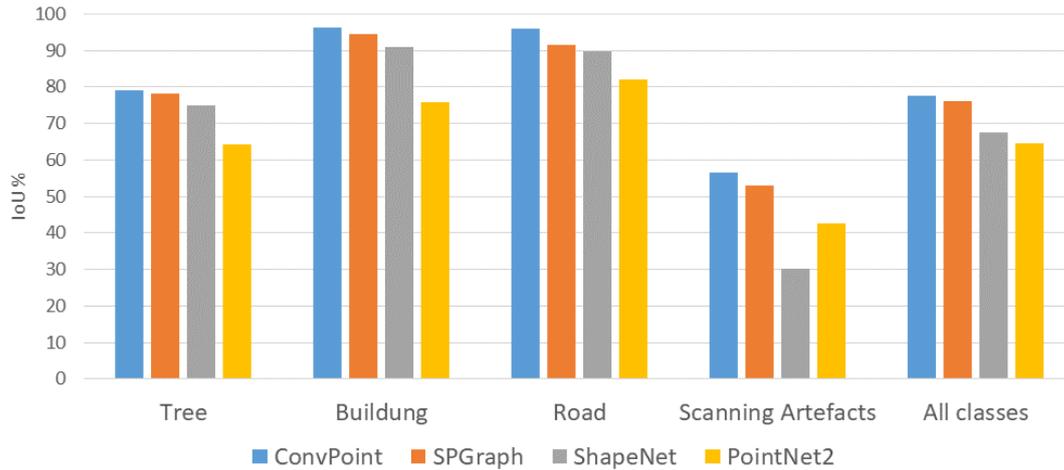


Figure 34: Semantic segmentation accuracy (IoU) of four common network architectures for the dataset: *Semantic3d.net* [40]. Selection of four from eight classes of this dataset. The class *Scanning Artefacts*, which is equal to the class *Erroneous point*, can be detected poorly compared to the larger classes. Values are taken from the leader board of [40].

The observations indicate that such influences exist (Figure 34). In PAPER 1, the influence of the presence or absence of the class *Erroneous points* are investigated. Figure 35a shows the results of the semantic segmentation without *Erroneous points*. The frequent classes *Tree* and *Building* are determined with more than 80% *recall* and *precision*. The infrequent class *Street Furniture* is determined with less than 10% *recall* and *precision*. If the class *Erroneous points* is added, *recall* and *precision* for all classes are lower than 54%, as shown in Figure 35b. From this example, it can be seen that there is an influence of the erroneous points in the semantic segmentation of point cloud datasets. Erroneous points are arranged similarly as object points, as erroneous points are caused by multiple and diffuse reflections. In the larger study of PAPER 2, the influence could be confirmed. However, with a large indoor datasets the influence is less. For infrequent class a positive effect of the presence of the class *Erroneous points* can also be observed by a higher *precision* value.

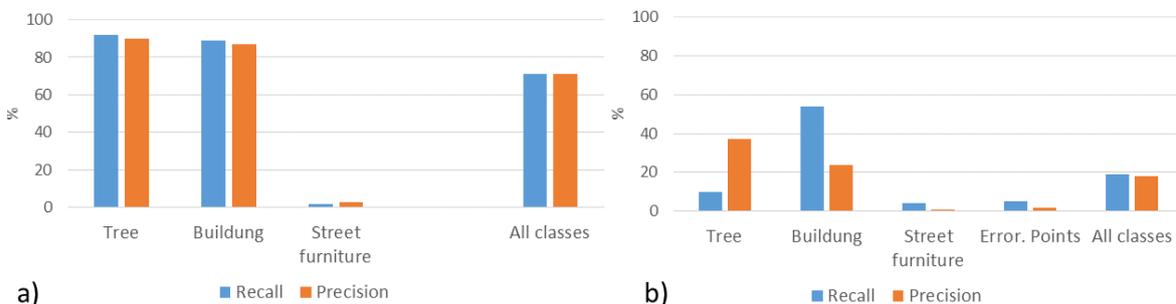


Figure 35: Comparison of semantic accuracy (*recall* and *precision*) on the point cloud of the HafenCity (outdoor) dataset: a) Without the class *Erroneous points* and b) With the class *Erroneous points*. Selection of three classes that have different frequencies in the dataset. Data from PAPER 1.

The influence of the class *Erroneous points* is therefore not the only crucial factor for the semantic accuracy, but the combination of the semantic classes and its point distribution. The division according to classes takes place on the basis of the class definition, that rules which classes are determined with the semantic segmentation. The best possible differentiation is always possible if the features of the point clouds can be clearly separated from each other. The ceiling and the floor can be well separated by different values for the feature variable height. Such considerations can be taken into account when developing the class definition. If, due to the task, a separation by classes with very similar features is not possible, a step-wise semantic segmentation can be performed as outlined in Figure 36. Similar classes are combined in a super class in the first stage (*Network A*) and then *Network B* is used for the separation. The influence of a class definition and the hierarchical process could be demonstrated in the study of PAPER 2. This shows a slight increase in semantic accuracy for the *Window* and *Door* classes. However, this developed process is strongly linked to the individual rooms.

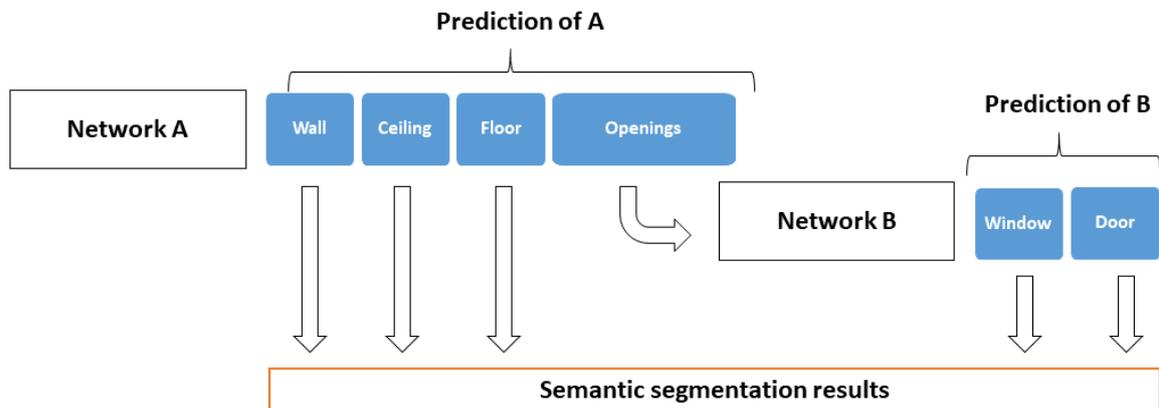


Figure 36: Step-wise semantic segmentation for improved differentiation of classes with similar features. With network A, a segmentation is performed for general classes, which is refined in network B.

An adjustment of the class definition does not necessarily lead to the classes having the same number of points. Classes such as *Wall*, *Floor* and *Ceiling* are more frequently represented classes in the point cloud, than *Doors*, *Furniture* and *Windows* due to their larger surfaces. The learning algorithm will learn these classes more often than the infrequent ones due to the more frequent feeding with points whose class is *wall*, *floor* or *ceiling*. To enhance learning in favor of the infrequent classes, their proportion can be artificially increased (Figure 37a), the inputs can be emphasized with a higher proportion of infrequent points (Figure 37c), or in a *loss* calculation, the points of the infrequent classes can be rewarded by a higher weight (Figure 37b). Extensions of points can be done randomly or by considering local conditions, as with the SMOTE method. These three approaches are investigated in PAPER 3 in several variants using the general HP set from RQ 3.2 (section 4.3.2). Again, a modest and scene-dependent increase in semantic accuracy is observed due to a higher *recall* for the infrequent classes.

4. Evaluation of the research results

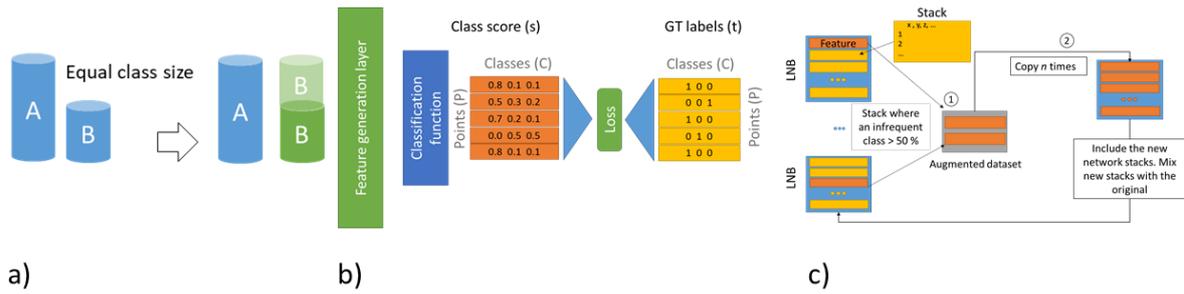


Figure 37: Dataset optimization methods for semantic point cloud segmentation: a) Dataset expansion by randomly copying points, b) weighting the *loss* function, and c) dataset expansion by copying inputs with infrequent points.

Conclusion and outlook: DHPs represent a measurable influence for the semantic segmentation of point clouds. The DHPs: Class definitions, proportion of erroneous points and class size differences influence the semantic segmentation results and are optimized in the context of this work. However, this optimization can only be valid for a proportion of the scenes. Further investigations on DHPs are necessary to establish rules for the optimal choice of them. An analysis of the semantics in the scenes is necessary.

5 Conclusion and outlook

This section summarizes the key findings, the conclusions and the responses to the RQs (section 5.1). Intermediate conclusions are summarized in section 4 at the end of each RQ. Next steps for further development and optimization based on the results of this work are described in section 5.2.

5.1 Conclusion

This thesis shows that the development of a workflow with which any type of point clouds can be semantically augmented, in any kind of application is not possible at the current state of the art. The key reasons are lack of knowledge about datasets and undefined rules for HPs. Nevertheless, DL methods are most suitable for semantic segmentation.

The semantic enhancement of 3D point clouds is a necessary step in order to produce highly accurate digital models of the real world. The semantics of point clouds is central for the usability and the interpretability, if automatic digital processing should or must be used. DL algorithms, such as point-based CNN, produce accurate semantic point clouds if optimal HPs and sufficient training point clouds are used. Optimal HPs, algorithms, and training data were explored on the *HCU main building* dataset and predominantly with the *PointNet* method in this thesis. In order to investigate the influences three developments were necessary:

- The development of a workflow for automatic semantic segmentation.
- The development of a tool for manual semantic segmentation.
- The development of a quality model for the process and the semantic point cloud.

The first challenge in workflow development was, that most DL methods use their own data pre-processing methods. This is dictated by the data format of the recording sensor and is not adapted for the optimal performance of the algorithm or to the data content.

The second challenge is the advancement of the hardware, APIs, and DL methods. To effectively consider new hardware and API developments as well as different DL algorithms, a modular workflow which is independent of the dataset formats has been developed. This workflow consists of the modules for feature extension, feature value normalization, input formatting, training, evaluation and application. Each module can be modified by a few parameters. The described workflow is a bridge between high-end developments and practical measurements.

Tools for point cloud annotations shall accelerate, simplify, standardize and optimize the very labor-intensive, individual process of point cloud annotation. The semantic point clouds are crucial, because they are the knowledge carrier of DL algorithms. The development of the PCCT is based on the above requirements and reduces individual human influences by

means of automatic segmentation. A reduction of the processing time compared to *Recap* was possible by 42% on average. However, this results in a decrease in semantic accuracy of up to 16% for *recall* and up to 12% for *precision* across all classes. The PCCT multi-user capability makes it ideal for studies in which different segmentation performances are investigated as an influence or allow efficient processing of large datasets by different users.

Which metrics are used for an evaluation is not standardized in literature and the exact expresses of each metric is sometimes not clear. In addition, these metrics usually only represent the semantic accuracy in relation to a GT dataset. All these ambiguities in the definitions limit the meaningfulness of the metrics. A complete evaluation of a semantic point cloud includes several characteristics, such as geometric accuracy, reliability, completeness, availability, and integrity. These characteristics of the semantic point cloud are represented by the quality model developed in this work. A complete evaluation and comparison of the dataset characteristics and performance of all processing steps is thus possible. The metrics are integrated into the quality model as quality parameters and define together with additional quality parameters a higher significance model for systematic investigations, comparisons and the examination for the suitability of data and algorithms in a specific application.

The three developments of the thesis are used to study the influence of datasets and point clouds in manual and automatic semantic segmentations. Differences in accuracy, effectiveness, and efficiency in manual semantic segmentation were identified, caused by the functions in the tools, the user training, and the point clouds. It was found that further developments of annotation tools are mandatory in order to produce sufficient training data for productive applications of DL algorithms. Training data point clouds are key materials, but have been rarely studied.

This work focuses on the characteristics of the datasets and the point clouds. The presence of erroneous points affects the semantic segmentation by decreasing the semantic accuracy of frequent classes. In contrast, for infrequent classes an increase in semantic accuracy is observed of up to 22% (interior) for the *PointNet* baseline method, if the class *Erroneous points* is part of the class definition. Unequal class partitioning leads to the fact that infrequent classes have a lower accuracy. In many examples of this thesis it can be observed that frequent classes are learned very well ($> 90\%$ *recall*) and infrequent classes are not learned ($< 50\%$ *recall*). Systematic and artificial modification of the point cloud dataset can improve the recognizability of infrequent classes (*recall* $> 50\%$). Classes that show very similar features are more difficult to separate than classes that have different features (e.g., heights). A hierarchical approach for the class definition could in some cases (e.g., windows and doors as openings) improve the semantic segmentation. This can be observed from the *recall*, which is up to 43% higher for the class *Openings* in the baseline method.

Finally, not only the characteristics of the individual points are crucial, but also the characteristics of a neighborhood. How the local and global neighborhood can be taken into account is discussed in many papers, but general rules that allow to apply them are not available yet.

Some attempts have been made to take into account different densities and different large areas in the input to the algorithm. However, its influence remains to be investigated in detail. Some possible approaches have been examined in this work and some new approaches will be explained in the outlook (section 5.2).

5.2 Outlook

The research of this dissertation reveals three additional research areas that require further investigations and developments. These research areas are the optimization of the input to DL algorithms (section 5.2.1), the manual pre-selection of feature variables (section 5.2.2), and the application of the quality model (section 5.2.3). Furthermore, research on algorithms, on strategies for optimization of HPs and combinations of ML and DL are the current issues.

5.2.1 Input format

In the semantic segmentation of point clouds with DL methods, the semantics are learned from the arrangement of the points and its additional feature variables, such as intensity, point normals or color values. Point clouds containing hundreds of thousands of points cannot be fed into a network architecture at once, so only a subset can be processed at each time. As a result, a subset of the information can be used for global feature extraction and classification. Information from the entire point cloud is not known at all (in the case of *PointNet*) or only insufficiently known (in the case of *RandLaNet* or *PointNet++*). For *PointNet* and 2D CNNs a possible approach to address this issue is the use of graphs as input format, such as described in section 2.6.2. Supplementing these methods of the literature, local and global *adjacency matrices* expressing the adjacency of the points can be computed from a kNN graph for each point. *Adjacency matrices* have the advantage that they order the topological relationships and can be represented in a 2D format. By multiplying the *adjacency matrices* with the features of the points, a tensor can be computed for input to a 2D CNN, such as *U-Net* (Figure 38). Also, the *adjacency matrix* for a local area may be used as a direct input to *PointNet* and is a carrier of additional information about the local relationships of the points.

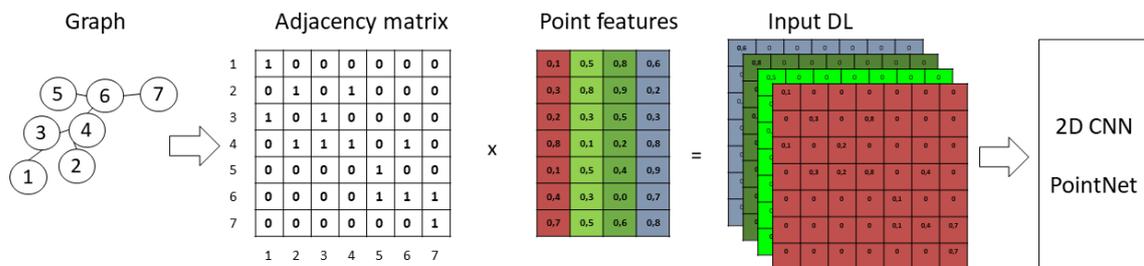


Figure 38: Process for creating an *adjacency matrix* and applying it as a network input.

Initial tests of this method show that an increase in semantic accuracy is possible with a *PointNet* architecture. Currently, data preparation is the bottleneck of this method. Larger studies with different datasets need to be conducted to validate this observation. The input format of the point cloud to the algorithm is seen as an important influence that must be investigated systematically in future research.

5.2.2 Hand-crafted feature selection

The manual selection of features for semantic segmentation with ML methods is necessary for pre-processing of point cloud data, which was investigated and optimized in [185]. This idea is applied to CNN in [40, 278] by computing *moments* and *eigenvalues* as additional features. Subsequently, an optimization of the set of input features is performed. *Eigenvalues* and *moments* carry information not only about the point itself, but also about the neighborhood, so they bring in more global information into the algorithm. To compute this type of features, it is necessary to define a local neighborhood over which the *eigenvalues* are determined. In a test, the *sum of eigenvalue*, the *planarity* and the *linearity* are calculated, using a radius of 3.5 cm for including the neighborhood. These differences of values are shown in Figures 39b to 39d. Figure 39a shows the GT classification of the point cloud. It can be seen that object boundaries can be distinguished more accurate than in the case of most spectral features.

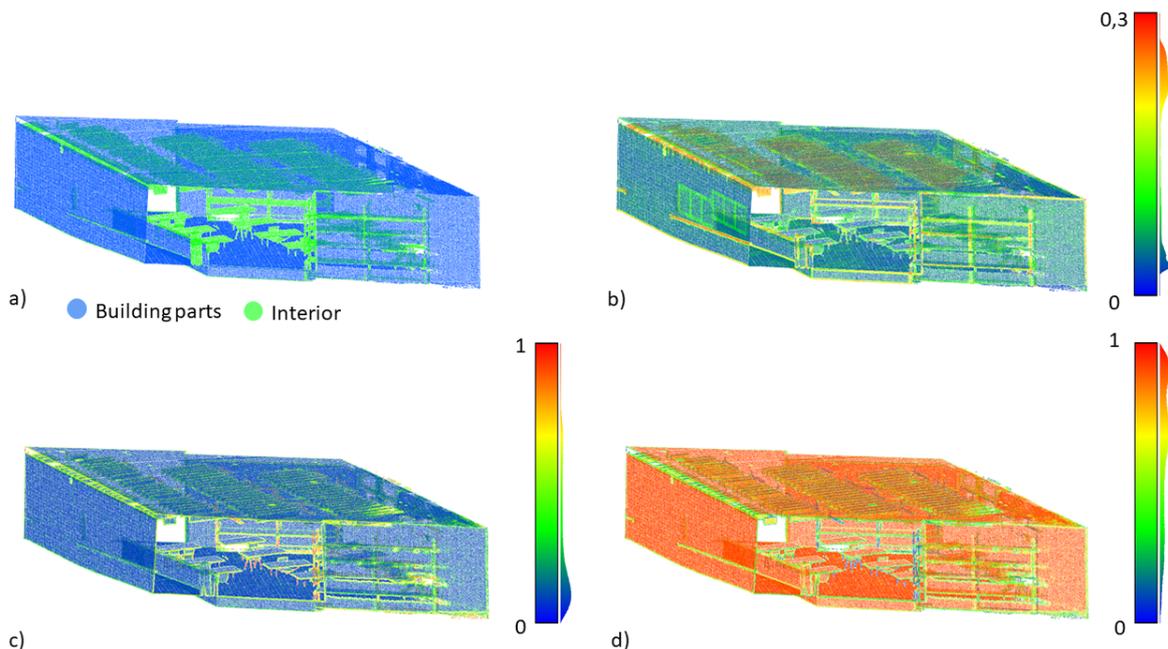


Figure 39: *Eigenvalue* based features calculated from geometric features (x, y, z): a) GT semantic segmentation. b) *Sum of eigenvalues* as a feature. c) *Planarity* as a feature. d) *Linearity* as a feature. A histogram is shown next to the legend.

In the experiment for this approach two tests were performed with the *PointNet* workflow. These experiments show that for the class combination *Erroneous points* and *Objects com-*

parable accuracies are achieved with baseline method (Figure 40). For the class combination of *Interior* and *Building parts*, it can be demonstrate that with the new *eigenvalue* feature set *recall* and *precision* decrease of more than 30% (Figure 41). In further studies the feature calculation and the variable selection have to be optimized to improve automatic semantic segmentation.

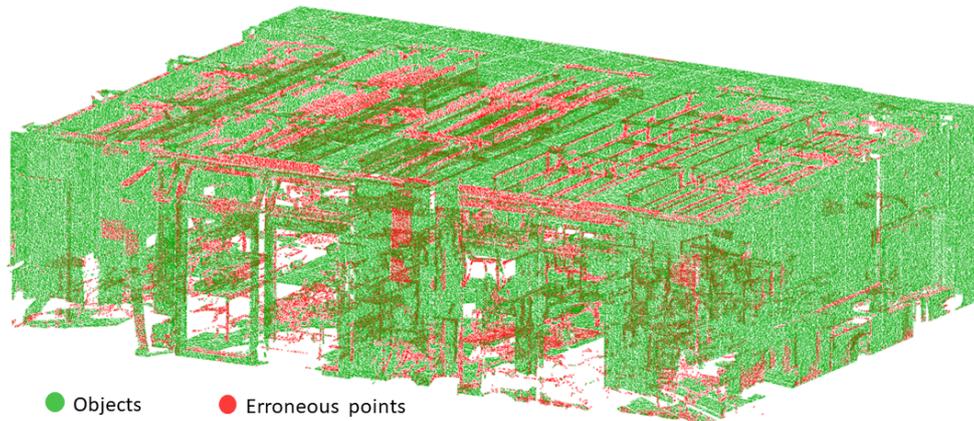


Figure 40: Semantic point cloud for the classes *Objects* and *Erroneous points*. The semantic segmentation is performed using the *PointNet*-based workflow with the features: *x-, y-, z-coordinates, sum of eigenvalues, planarity* and *linearity*.

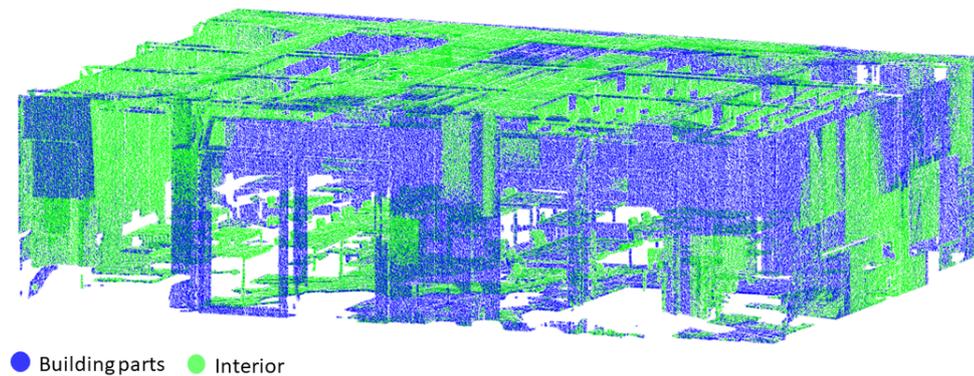


Figure 41: Semantic point cloud for the classes *Building parts* and *Interior*. The semantic segmentation is performed using the *PointNet*-based workflow with the features: *x-, y-, z-coordinates, sum of eigenvalues, planarity* and *linearity*.

5.2.3 Point cloud quality assessment

The evaluation of semantic segmentation in real-world applications has been treated only marginally in the developments so far. Since the influence of the data and its enhancement is crucial for the performance of the algorithms, these should be investigated in more detail. The developed quality model of this work provides a basis for this, for which thresholds per quality parameter still have to be defined. These should be primary for frequent applications of semantic segmentation, as it is the case for indoor scenes, facades, or street scenes. In addition, these thresholds must be algorithm-dependently determined.

Bibliography

- [1] P. Welchering. *Nach der Corona-Pandemie - Wie die Digitalisierung der Arbeitswelt jetzt weitergeht*. Ed. by Deutschlandfunk. 2021. URL: <https://www.deutschlandfunk.de/nach-der-corona-pandemie-wie-die-digitalisierung-der-100.html> (visited on 04/15/2023).
- [2] H. Kahmen. *Angewandte Geodäsie: Vermessungskunde*. De Gruyter Verlag, 2005. DOI: 10.1515/9783110911145.
- [3] A. Wieser, H. Kuhlmann, V. Schwieger, and W. Niemeier. "Ingenieurgeodäsie – eine Einführung". In: *Ingenieurgeodäsie*. Ed. by W. Schwarz. Springer Spektrum, Berlin, Heidelberg, 2017, pp. 1–22. DOI: 10.1007/978-3-662-47188-3_19.
- [4] V. Stojanovic, H. Shoushtari, C. Askar, A. Scheider, C. Schuldt, N. Hellweg, and H. Sternberg. "A Conceptual Digital Twin for 5G Indoor Navigation". In: *The Eleventh International Conference on Mobile Services, Resources and Users - MOBILITY 2021*. 2021, pp. 5–14.
- [5] H. Hosamo, M. H. Hosamo, H. K. Nielsen, P. R. Svennevig, and K. Svidt. "Digital Twin of HVAC system (HVACDT) for multiobjective optimization of energy consumption and thermal comfort based on BIM framework with ANN-MOGA". In: *Advances in Building Energy Research* 17.2 (2022), pp. 125–171. DOI: 10.1080/17512549.2022.2136240.
- [6] R. Kaden, C. Clemen, R. Seuß, J. Blankenbach, R. Becker, A. Eichhorn, A. Donaubauer, and U. Gruber. *Leitfaden Geodäsie und BIM*. Tech. rep. 2.1. DVW e.V. und Runder Tisch GIS e.V., 2020. URL: <https://dvw.de/images/anhang/2757/leitfaden-geodaesie-und-bim2020onlineversion.pdf> (visited on 07/01/2022).
- [7] T. Bender, M. Härtig, E. Jaspers, M. Krämer, M. May, M. Schlundt, and N. Turianskyj. "Building Information Modeling". In: *CAFM-Handbuch*. Springer Fachmedien Wiesbaden, 2018. Chap. 11, pp. 295–324. DOI: 10.1007/978-3-658-21357-2_11.
- [8] BIM.Hamburg. *BIM-Leitfaden für die FHH Hamburg*. Tech. rep. BIM.Hamburg, 2019.
- [9] DIN 18710. *Engineering survey*. Deutsche Norm, 2010.
- [10] R. Becker, E. Lublasser, J. Martens, R. Wollenberg, H. Zhang, S. Brell-Cokcan, and J. Blankenbach. "Enabling BIM for Property Management of Existing Buildings Based on Automated As-is Capturing". In: *Proceedings of the International Symposium on Automation and Robotics in Construction*. International Association for Automation and Robotics in Construction, 2019. DOI: 10.22260/isarc2019/0028.
- [11] K. Soliman, K. Naji, M. Gunduz, O. B. Tokdemir, F. Faqih, and T. Zayed. "BIM-based Facility Management Models for Existing Buildings". In: *Journal of Engineering Research* (2021). DOI: 10.36909/jer.11433.

- [12] N. Hellweg, C. Schuldt, H. Shoushtari, and H. Sternberg. "Potenziale für Anwendungsfälle des Facility Managements von Gebäuden durch die Nutzung von Bauwerksinformationsmodellen als Datengrundlage für Location-Based Services im 5G-Netz". In: *21. Internationale Geodätische Woche, Obergurgl*. Wichmann Herbert, 2021. ISBN: 3879077029.
- [13] Y. Li, Y. Zhang, X. Pan, and J. E. Taylor. "BIM-based determination of indoor navigation sign layout using hybrid simulation and optimization". In: *Automation in Construction* 139 (2022). ISSN: 0926-5805. DOI: 10.1016/j.autcon.2022.104243.
- [14] N. Hellweg, C. Schuldt, H. Shoushtari, J. Müller-Lietzkow, and H. Sternberg. "5G based Indoor Navigation at Hafencity University Hamburg". In: *27th World Congress on Intelligent Transport Systems*. Hamburg, Germany, 2021.
- [15] H. Alavi, N. Forcada, S.-L. Fan, and W. San. "BIM-based Augmented Reality for Facility Maintenance Management". In: *Proceedings of the 2021 European Conference on Computing in Construction*. University College Dublin, 2021. DOI: 10.35490/ec3.2021.180.
- [16] P. Boguslawski, S. Zlatanova, D. Gotlib, M. Wyszomirski, M. Gnat, and P. Grzempowski. "3D Building Interior Modelling for Navigation in Emergency Response Applications". In: *International Journal of Applied Earth Observation and Geoinformation* 114 (2022), p. 103066. DOI: 10.1016/j.jag.2022.103066.
- [17] S. Nikoohemat, P. Godoy, N. Valkhoff, M. W. van Leeuwen, R. Voûte, and V. V. Lehtola. "Point Cloud Based 3d Models for Agent Based Simulations in Social Distancing and Evacuation". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-4-2021* (2021), pp. 113–120. DOI: 10.5194/isprs-annals-v-4-2021-113-2021.
- [18] S. DeGeyter, J. Vermandere, H. D. Winter, M. Bassier, and M. Vergauwen. "Point Cloud Validation: On the Impact of Laser Scanning Technologies on the Semantic Segmentation for BIM Modeling and Evaluation". In: *Remote Sensing* 14.3 (2022), p. 582. DOI: 10.3390/rs14030582.
- [19] Y. Pan, A. Braun, A. Borrmann, and I. Brilakis. "Void-Growing: A Novel Scan-To-BIM Method for Manhattan World Buildings from Point Cloud". In: *Proceedings of the 2021 European Conference on Computing in Construction*. University College Dublin, 2021, pp. 312–321. DOI: 10.35490/EC3.2021.162.
- [20] A. Nüchter, D. Borrmann, P. Koch, M. Kühn, and S. May. "A Man-Portable, Imu-Free Mobile Mapping System". In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences II-3/W5* (2015), pp. 17–23. DOI: 10.5194/isprsannals-ii-3-w5-17-2015.
- [21] C. Wang, B. Samari, and K. Siddiqi. "Local Spectral Graph Convolution for Point Set Feature Learning". In: *Computer Vision – ECCV* (2018), pp. 56–71. ISSN: 0302-9743. DOI: 10.1007/978-3-030-01225-0_4.

- [22] F. Keller. "Entwicklung eines forschungsorientierten Multi-Sensor-System zum kinematischen Laserscannings innerhalb von Gebäuden". PhD thesis. HafenCity University Hamburg, 2015.
- [23] T. Lovas, K. Hadzijanisz, V. Papp, and A. J. Somogyi. "Indoor Building Survey Assessment". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B1-2020* (2020), pp. 251–257. DOI: 10.5194/isprs-archives-xliii-b1-2020-251-2020.
- [24] G. Lim and N. Doh. "Automatic Reconstruction of Multi-Level Indoor Spaces from Point Cloud and Trajectory". In: *Sensors* 21.10 (2021), p. 3493. DOI: 10.3390/s21103493.
- [25] C. Thomson and J. Boehm. "Indoor Modelling Benchmark for 3D Geometry Extraction". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-5* (2014), pp. 581–587. DOI: 10.5194/isprsarchives-xl-5-581-2014.
- [26] M. Previtali, L. Barazzetti, R. Brumana, and M. Scaioni. "Towards Automatic Indoor Reconstruction of Cluttered Building Rooms from Point Clouds". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-5* (2014), pp. 281–288. DOI: 10.5194/isprsannals-ii-5-281-2014.
- [27] C. Askar, E. Barnefske, N. Hellweg, V. Stojanovic, O. Konkova, and H. Sternberg. "Semantically-Rich Floorplans from Indoor Point Clouds". In: *20. Ingenieurvermessung 2023 in Zurich*. 2023.
- [28] P. Hübner, M. Weinmann, S. Wursthorn, and S. Hinz. "Automatic Voxel-Based 3D Indoor Reconstruction and Room Partitioning from Triangle Meshes". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 181 (2021), pp. 254–278. DOI: 10.1016/j.isprsjprs.2021.07.002.
- [29] T. H. Kolbe, T. Kutzner, C. S. Smyth, C. Nagel, C. Roensdorf, and C. Heazel. *OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard*. Tech. rep. 20-010. Open Geospatial Consortium, Arlington, VA, USA, 2021. eprint: <https://docs.ogc.org/is/20-010/20-010.pdf>. (Visited on 04/14/2023).
- [30] T. Gilbert, C. Rönsdorf, J. Plume, S. Simmons, N. Nisbet, H.-C. Gruler, T. H. Kolbe, L. van Berlo, and A. Mercer. *Built Environment Data Standards and their Integration: An Analysis of IFC, CityGML and Landinfra*. Tech. rep. 19-091r1. Open Geospatial Consortium & buildingSMART International, 2020, pp. 1–16.
- [31] OpenGeospatialConsortium. *IndoorGML Standard, Version: 1.1*. 2023. URL: <https://www.ogc.org/standard/indoorgml/> (visited on 04/14/2023).
- [32] buildingSMART. *Industry Foundation Classes 4.0.2.1*. 2021. URL: <https://standards.buildingsmart.org> (visited on 06/24/2021).

- [33] B. Plaß and T. Klauer. “Next Generation Scan-to-BIM: Ein neuer Ansatz zur strukturierten Datenerfassung für as-built Indoor Modelle”. In: *Leitfaden Geodäsie und BIM*. 2020, pp. 176–178.
- [34] E. S. Malinverni, R. Pierdicca, M. Paolanti, M. Martini, C. Morbidoni, F. Matrone, and A. Lingua. “Deep Learning for Semantic Segmentation of 3D Point Cloud”. In: *ISPRS - The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W15* (2019), pp. 735–742. ISSN: 2194-9034. DOI: 10.5194/isprs-archives-xlii-2-w15-735-2019.
- [35] F. Noichl, A. Braun, and A. Borrmann. ““BIM-to-Scan” for Scan-to-BIM: Generating Realistic Synthetic Ground Truth Point Clouds based on Industrial 3D Models”. In: *Proceedings of the 2021 European Conference on Computing in Construction*. University College Dublin, 2021, pp. 164–172. DOI: 10.35490/ec3.2021.166.
- [36] G. Cai and Y. Pan. “Understanding the Imperfection of 3D point Cloud and Semantic Segmentation Algorithms for 3D Models of Indoor Environment”. In: *AGILE: GI-Science Series 3* (2022), pp. 1–10. DOI: 10.5194/agile-giss-3-2-2022.
- [37] L. Winiwarter and G. Mandlbürger. “Classification of 3D Point Clouds using Deep Neural Networks”. In: *Dreiländertagung der DGPF, der OVG und der SGPF in Wien, Österreich – Publikationen der DGPF*. Vol. 28. 2019, pp. 663–674.
- [38] L. Winiwarter, G. Mandlbürger, and N. Pfeifer. “Klassifizierung von 3D ALS Punktwolken mit Neuronalen Netzen”. In: *20. Internationale Geodätische Woche Oberurgl*. Vol. 20. 2019, pp. 254–273.
- [39] 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 IV-1/W1* (2017), pp. 157–164. DOI: 10.5194/isprs-annals-iv-1-w1-157-2017.
- [40] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys. “Semantic3d.net: A New Large-scale Point Cloud Classification Benchmark”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-1-W1* (2017), pp. 91–98. DOI: 10.5194/isprs-annals-iv-1-w1-91-2017.
- [41] T. Hackel, J. D. W. Wegner, and K. Schindler. “Fast Semantic Segmentation of 3D Point Clouds with Strongly Varying Densit.” In: *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences 3.3* (2016). DOI: 10.5194/isprsannals-iii-3-177-2016.
- [42] M. Soilán, R. Lindenbergh, B. Riveiro, and A. Sánchez-Rodríguez. “PointNet for the Automatic Classification of Aerial Point Clouds”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-2/W5* (2019), pp. 445–452. DOI: 10.5194/isprs-annals-iv-2-w5-445-2019.

- [43] Y. Lin, G. Vosselman, Y. Cao, and M. Y. Yang. "Active and Incremental Learning for Semantic ALS Point Cloud Segmentation". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 169 (2020), pp. 73–92. DOI: 10.1016/j.isprsjprs.2020.09.003.
- [44] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. "RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 11105–11114. DOI: 10.1109/cvpr42600.2020.01112.
- [45] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 77–85. DOI: 10.1109/cvpr.2017.16.
- [46] A. Geiger, P. Lenz, and R. Urtasun. "Are We Ready for Autonomous Driving? The KITTI and Vision Benchmark and Suite." In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361. DOI: 10.1109/cvpr.2012.6248074.
- [47] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. "Vision meets Robotics: The KITTI Dataset". In: *International Journal of Robotics Research (IJRR)* (2013). DOI: 10.1177/0278364913491297.
- [48] A. Serna, B. Marcotegui, F. Goulette, and J.-E. Deschaud. "Paris-Rue-Madame Database - A 3D Mobile Laser Scanner Dataset for Benchmarking Urban Detection, Segmentation and Classification Methods". In: *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods*. SCITEPRESS - Science, 2014. DOI: 10.5220/0004934808190824.
- [49] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. "KPConv: Flexible and Deformable Convolution for Point Clouds". In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2019. DOI: 10.1109/iccv.2019.00651.
- [50] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang. "3D Recurrent Neural Networks with Context Fusion for Point Cloud Semantic Segmentation". In: *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Springer International Publishing, 2018, pp. 415–430. DOI: 10.1007/978-3-030-01234-2_25.
- [51] D. Bazazian and D. Nahata. "DCG-Net: Dynamic Capsule Graph Convolutional Network for Point Clouds". In: *IEEE Access* 8 (2020), pp. 188056–188067. DOI: 10.1109/access.2020.3031812.
- [52] S. Son, G. Lee, J. Jung, J. Kim, and K. Jeon. "Automated Generation of a Model View Definition from an Information Delivery Manual Using idmXSD and buildingSMART Data Dictionary". In: *Advanced Engineering Informatics* 54 (2022), p. 101731. ISSN: 1474-0346. DOI: 10.1016/j.aei.2022.101731.

- [53] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences”. In: *International Conference on Computer Vision*. IEEE, 2019, pp. 9296–9306. DOI: 10.1109/ICCV.2019.00939.
- [54] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss. “Towards 3D LiDAR-Based Semantic Scene Understanding of 3d Point Cloud Sequences: The SemanticKITTI Dataset”. In: *The International Journal of Robotics Research* 40.8-9 (2021), pp. 959–967. DOI: 10.1177/02783649211006735.
- [55] M. Ibrahim, N. Akhtar, M. Wise, and A. Mian. “Annotation Tool and Urban Dataset for 3D Point Cloud Semantic Segmentation”. In: *IEEE Access* 9 (2021), pp. 35984–35996. DOI: 10.1109/access.2021.3062547.
- [56] R. Huang, Z. Ye, D. Hong, Y. Xu, and U. Stilla. “Semantic Labeling and Refinement of Lidar Point Clouds Using Deep Neural Network in Urban Areas”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-2/W7* (2019), pp. 63–70. DOI: 10.5194/isprs-annals-iv-2-w7-63-2019.
- [57] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. “Conditional Random Fields as Recurrent Neural Networks”. In: *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2015, pp. 1529–1537. DOI: 10.1109/iccv.2015.179.
- [58] 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”. In: *International Conference on Robotics and Automation*. IEEE, 2018. DOI: 10.1109/icra.2018.8462926.
- [59] K. A. Ogori, A. Diakit , T. Krijnen, H. Ledoux, and J. Stoter. “Processing BIM and GIS Models in Practice: Experiences and Recommendations from a GeoBIM Project in the Netherlands”. In: *ISPRS International Journal of Geo-Information* 7.8 (2018), p. 311. DOI: 10.3390/ijgi7080311.
- [60] P. Dorninger and N. Pfeifer. “A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction and Regularization from Airborne Laser Scanning Point Clouds”. In: *Sensors* 8.11 (2008), pp. 7232–7343. DOI: 10.3390/s8117323.
- [61] M. Petry, T. Becker, U. Adreanx, and D. Raue. “Vermessungsarbeiten im Katastrophengebiet Ahrtal – Bestandserfassung mittels UAV und Scanner, Fusion der Daten und Dokumentation der neuen Ersatzbr cken”. In: *22. Internationale Geod tische Woche Oberurgl*. Ed. by T. Weinold. 2023, pp. 67–77. ISBN: 978-3-87907-738-0.
- [62] E. Barnefske and H. Sternberg. “Automatisch semantisch-segmentierte Punktwolken – M glichkeiten und Herausforderungen”. In: *DVW-Seminar MST 2022 – von (A)nwendungen bis (Z)ukunftstechnologien. DVW-Schriftenreihe*. Vol. 103. Wi sner-Verlag, Augsburg, 2022, pp. 173–184. ISBN: 978-3-95786-322-5.

- [63] T. Luhmann. *Nahbereichsphotogrammetrie Grundlagen - Methoden - Beispiele*. Wichmann Verlag, Berlin, Offenbach, 2018. ISBN: 9783879076413.
- [64] Y. Xu and U. Stilla. “Toward Building and Civil Infrastructure Reconstruction From Point Clouds: A Review on Data and Key Techniques”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), pp. 2857–2885. DOI: 10.1109/jstars.2021.3060568.
- [65] F. Remondino, M. G. Spera, E. Nocerino, F. Menna, and F. Nex. “State of the Art in High Density Image Matching”. In: *The Photogrammetric Record* 29.146 (2014), pp. 144–166. DOI: 10.1111/phor.12063.
- [66] Y. Han, W. Liu, X. Huang, S. Wang, and R. Qin. “Stereo Dense Image Matching by Adaptive Fusion of Multiple-Window Matching Results”. In: *Remote Sensing* 12.19 (2020), p. 3138. DOI: 10.3390/rs12193138.
- [67] R. Düvel. “Entwicklung eines Photogrammetrischen Aufnahmekonzeptes für die Bauwerkspüfung von Tunneln sowie eines Auswerteprozesses mit einer Gopro-Kamera und einer Sony Alpha R III”. Bachelorthesis. HafenCity University Hamburg, 2021.
- [68] S. DeGeyter, M. Bassier, and M. Vergauwen. “Automated Training Data Creation for Semantic Segmentation of 3D Point Clouds”. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVI-5/W1-2022* (2022), pp. 59–67. DOI: 10.5194/isprs-archives-xlvi-5-w1-2022-59-2022.
- [69] H. Sarbolandi, D. Lefloch, and A. Kolb. “Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect”. In: *Computer Vision and Image Understanding* 139 (2015), pp. 1–20. DOI: 10.1016/j.cviu.2015.05.006.
- [70] R. Shults, E. Levin, R. Habibi, S. Shenoy, O. Honcheruk, T. Hart, and Z. An. “Capability of Matterport 3D Camera for Industria Archaeolog Sites Inventory”. In: *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W11* (2019), pp. 1059–1064. DOI: 10.5194/isprs-archives-xlii-2-w11-1059-2019.
- [71] S. A. Bello, S. Yu, and C. Wang. “Review: Deep Learning on 3D Point Clouds”. In: *Remote Sensing* 12.11 (2020), p. 1729. DOI: 10.3390/rs12111729.
- [72] Y. Xie, J. Tian, and X. X. Zhu. “Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation”. In: *IEEE Geoscience and Remote Sensing Magazine* 8.4 (2020), pp. 38–59. DOI: 10.1109/mgrs.2019.2937630.
- [73] C. Mahn. “Entwicklung und Evaluierung eines ROS-basierten Innenraumerfassungssystem mit der Tiefenkamera Intel Realsense D435i”. Bachelorthesis. HafenCity University Hamburg, 2022.
- [74] M. Tölgyessy, M. Dekan, L. Chovanec, and P. Hubinský. “Evaluation of the Azure Kinect and its Comparison to Kinect V1 and Kinect V2”. In: *Sensors* 21.2 (2021), p. 413. DOI: 10.3390/s21020413.

- [75] O. Wasenmüller and D. Stricker. “Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision”. In: *Computer Vision – ACCV 2016 Workshops*. Springer International Publishing, 2017, pp. 34–45. DOI: 10.1007/978-3-319-54427-4_3.
- [76] K. Sauermann. “Modellierung von Bestandsbauwerken am Beispiel der Messe Dortmund, Halle 4”. In: *22. Internationale Geodätische Woche Obergurgl*. Ed. by T. Weinold. 2023, pp. 67–77. ISBN: 978-3-87907-738-0.
- [77] C. Wang, S. Hou, C. Wen, Z. Gong, Q. Li, X. Sun, and J. Li. “Semantic Line Framework-Based Indoor Building Modeling Using Backpacked Laser Scanning Point Cloud”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 143 (2018), pp. 150–166. DOI: 10.1016/j.isprsjprs.2018.03.025.
- [78] B. Stahl and A. Reiterer. “Mobile Mapping Platform with Integrated End-To-End Data Processing Chain for Smart City Applications”. In: *Remote Sensing Technologies and Applications in Urban Environments VII*. Ed. by N. Chrysoulakis, T. Erbertseder, and Y. Zhang. SPIE, 2022. DOI: 10.1117/12.2633965.
- [79] T. Luhmann, S. Robson, S. Kyle, and J. Boehm. *Close-Range Photogrammetry and 3D Imaging*. De Gruyter, 2013. DOI: 10.1515/9783110302783.
- [80] F. Neitzel, B. Gordon, and D. Wujanz. *DVW-Merkblatt 7-2014, Verfahren zur standardisierten Überprüfung von terrestrischen Laserscannern (TLS)*. Tech. rep. DVW, 2014. URL: <https://dvw.de/veroeffentlichungen/standpunkte/1149-verfahren-zur-standardisierten-ueberpruefung-von-terrestrischen-laserscannern-tls> (visited on 10/28/2021).
- [81] D. Wujanz, M. Burger, F. Tschirschwitz, T. Nietzschmann, F. Neitzel, and T. Kersten. “Determination of Intensity-Based Stochastic Models for Terrestrial Laser Scanners Utilising 3D-Point Clouds”. In: *Sensors* 18.7 (2018), p. 2187. DOI: 10.3390/s18072187.
- [82] H. Neuer. “Qualitätsbetrachtungen zu TLS-Daten”. In: *DVW-Seminar Qualitätssicherung geodätischer Mess- und Auswerteverfahren*. Vol. 95. DVW-Arbeitskreis 3 »Messmethoden und Systeme«, 2019, pp. 69–89.
- [83] T. P. Kersten, M. Lindstaedt, and M. Stange. “Geometrische Genauigkeitsuntersuchungen aktueller terrestrischer Laserscanner im Labor und im Feld”. In: *AVN* 2.128 (2021), pp. 59–67.
- [84] B. Ortner, S. Papp, and M. Meir-Huber. “Einleitung”. In: *Handbuch Data Science*. Carl Hanser Verlag GmbH & Co. KG, 2019, pp. 1–14. DOI: 10.3139/9783446459755.001.
- [85] T. A. Runkler. “Einführung”. In: *Data Mining*. Springer Fachmedien Wiesbaden, 2015, pp. 1–3. DOI: 10.1007/978-3-8348-2171-3_1.
- [86] G. Langs and R. Wazir. “Machine Learning”. In: *Handbuch Data Science*. Carl Hanser Verlag GmbH & Co. KG, 2019, pp. 177–198. DOI: 10.3139/9783446459755.006.

- [87] M. Meir-Huber, S. Papp, and B. Ortner. “Datenarchitekturen”. In: *Handbuch Data Science*. Carl Hanser Verlag GmbH & Co. KG, 2019, pp. 103–123. DOI: 10.3139/9783446459755.003.
- [88] T. Dullien. “Maschinelles Lernen und künstliche Intelligenz in der Informationssicherheit”. In: *Datenschutz und Datensicherheit* 42.10 (2018), pp. 618–622. DOI: 10.1007/s11623-018-1012-3.
- [89] R. Wazir and G. Langs. “Statistik-Grundlagen”. In: *Handbuch Data Science*. Carl Hanser Verlag GmbH & Co. KG, 2019, pp. 141–176. DOI: 10.3139/9783446459755.005.
- [90] A. Ng and K. Soo. “Das Wichtigste in Kürze ...” In: *Data Science – Was ist das eigentlich?!* Springer Berlin Heidelberg, 2018, pp. 1–18. DOI: 10.1007/978-3-662-56776-0_1.
- [91] B. Ortner. “Data Pipelines”. In: *Handbuch Data Science*. Carl Hanser Verlag GmbH & Co. KG, 2019, pp. 125–140. DOI: 10.3139/9783446459755.004.
- [92] T. Wiltsoch. “Sichere Information durch infrastrukturgestützte Fahrerassistenzsysteme zur Steigerung der Verkehrssicherheit an Straßenknotenpunkten”. PhD thesis. University Stuttgart, 2004.
- [93] T. A. Runkler. “Datenvorverarbeitung”. In: *Data Mining*. Springer Fachmedien Wiesbaden, 2015, pp. 23–36. DOI: 10.1007/978-3-8348-2171-3_3.
- [94] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet. “Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 105 (2015), pp. 286–304. DOI: 10.1016/j.isprsjprs.2015.01.016.
- [95] P. F. Felzenszwalb and D. P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *International Journal of Computer Vision* 59.2 (2004), pp. 167–181. DOI: 10.1023/b:visi.0000022288.19776.77.
- [96] R. Schnabel, R. Wahl, and R. Klein. “Efficient RANSAC and for Point-Cloud and Shape Detection”. In: *Computer Graphics Forum* 26.2 (2007), pp. 214–226. ISSN: 0167-7055. DOI: 10.1111/j.1467-8659.2007.01016.x.
- [97] M. A. Fischler and R. C. Bolles. “Random Sample Consensus”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395. DOI: 10.1145/358669.358692.
- [98] N. Shukla. *Machine Learning with Tensorflow*. Ed. by K. Fricklas. manning pub, 2018. 272 pp. ISBN: 1617293873.
- [99] A. Y. Ng and M. I. Jordan. “On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes.” In: *Advances in Neural Information Processing Systems* (2002), pp. 841–848.
- [100] E. Che, J. Jung, and M. Olsen. “Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art Review”. In: *Sensors* 19.4 (2019), p. 810. DOI: 10.3390/s19040810.

- [101] L. I. Smith. *A tutorial on Principal Components Analysis*. Tech. rep. Report No. OUCS-2002-12. University of Otago, NZ, 2002. URL: <http://hdl.handle.net/10523/7534> (visited on 02/21/2023).
- [102] T. A. Runkler. “Datenvisualisierung”. In: *Data Mining*. Springer Fachmedien Wiesbaden, 2015, pp. 37–58. DOI: 10.1007/978-3-8348-2171-3_4.
- [103] Y. Xu, L. Hoegner, S. Tuttas, and U. Stilla. “Voxel- and Graph-Based Point Cloud Segmentation of 3D Scenes Using Perceptual Grouping Laws”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-1/W1* (2017), pp. 43–50. DOI: 10.5194/isprs-annals-iv-1-w1-43-2017.
- [104] R. Triebel, J. Shin, and R. Y. Siegwart. “Segmentation and Unsupervised Part-based Discovery of Repetitive Objects”. In: *Robotics: Science and Systems VI*. Robotics: Science and Systems Foundation, 2010. DOI: 10.15607/RSS.2010.VI.009.
- [105] A.-V. Vo, L. Truong-Hong, D. F. Laefer, and M. Bertolotto. “Octree-Based Region Growing for Point Cloud Segmentation”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 104 (2015), pp. 88–100. DOI: 10.1016/j.isprsjprs.2015.01.011.
- [106] M. Ahmed, R. Seraj, and S. M. S. Islam. “The k-means Algorithm: A Comprehensive Survey and Performance Evaluation”. In: *Electronics* 9.8 (2020), p. 1295. DOI: 10.3390/electronics9081295.
- [107] H. Zhang. “The Optimality of Naive Bayes”. In: *The Florida AI Research Society*. 2004.
- [108] T. A. Runkler. “Klassifikation”. In: *Data Mining*. Springer Fachmedien Wiesbaden, 2015, pp. 89–107. DOI: 10.1007/978-3-8348-2171-3_8.
- [109] J. C. Stoltzfus. “Logistic Regression: A Brief Primer”. In: *Academic Emergency Medicine* 18.10 (2011), pp. 1099–1104. DOI: 10.1111/j.1553-2712.2011.01185.x.
- [110] L. Jiang, Z. Cai, D. Wang, and S. Jiang. “Survey of Improving K-Nearest-Neighbor for Classification”. In: *IEEE Fourth International Conference on Fuzzy Systems and Knowledge Discovery*. IEEE, 2007. DOI: 10.1109/fskd.2007.552.
- [111] G. Shakhnarovich and T. Darrell. *Nearest-Neighbor Methods in Learning and Vision. Theory and Practice (Neural Information Processing)*. The MIT Press, 2006, p. 262. ISBN: 9780262195478.
- [112] K. Kunze. *Hauptseminar Machine Learning: Support Vector Machines, Kernels*. 2004. URL: https://campar.in.tum.de/twiki/pub/Far/MachineLearningWiSe2003/kunze_ausarbeitung.pdf (visited on 01/12/2023).
- [113] S. Levine. *Week 6: Support Vector Machines*. 2016. URL: https://courses.cs.washington.edu/courses/cse446/16sp/svm_2.pdf (visited on 01/13/2023).
- [114] B. DeVille. “Decision Trees”. In: *Wiley Interdisciplinary Reviews: Computational Statistics* 5.6 (2013), pp. 448–455. DOI: 10.1002/wics.1278.
- [115] L. Breiman. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/a:1010933404324.

- [116] D. Koguciuk, Ł. Chechliński, and T. El-Gaaly. “3D Object Recognition with Ensemble Learning - A Study of Point Cloud-Based Deep Learning Models”. In: *Advances in Visual Computing* (2019), pp. 100–114. DOI: 10.1007/978-3-030-33723-0_9.
- [117] S. Strecker. “Künstliche Neuronale Netze - Aufbau und Funktionsweise”. In: *Arbeitspapiere WI*. Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes Gutenberg-Universität Mainz, 1997.
- [118] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016. 800 pp. ISBN: 0262035618. URL: https://www.ebook.de/de/product/26337726/ian_goodfellow_yoshua_bengio_aaron_courville_deep_learning.html (visited on 01/13/2023).
- [119] T. Rashid. *Neuronale Netze selbst programmieren*. D Punkt Verlag GmbH, 2017. 232 pp. ISBN: 3960090439.
- [120] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer. “Efficient Processing of Deep Neural Networks: A Tutorial and Survey”. In: *Proceedings of the IEEE* 105.12 (2017), pp. 2295–2329. DOI: 10.1109/jproc.2017.2761740.
- [121] M. Heinert. “Artificial Neural Networks – How to Open the Black Boxes?” In: *First Workshop on Application of Artificial Intelligence and Innovations in Engineering Geodesy*. 2008, pp. 42–62.
- [122] I. Goodfellow. *Deep Feedforward Networks*. 2016. URL: http://www.deeplearningbook.org/lecture_slides.html (visited on 01/13/2023).
- [123] L. Fei-Fei, J. Wu, and R. Gao. *CS231n: Convolutional Neural Networks for Visual Recognition*. Ed. by Stanford vision and Learning Lab. 2020. URL: <https://cs231n.github.io/neural-networks-case-study/> (visited on 02/02/2023).
- [124] D. Maturana and S. Scherer. “VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2015, pp. 922–928. DOI: 10.1109/iros.2015.7353481.
- [125] N. Höft. “Bildsegmentation in Objekt-Klassen mit Konvolutionalen Neuronalen Netzen”. Bachelorthesis. University Bonn, 2014.
- [126] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun. “Deep Learning for 3D Point Clouds: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.12 (2019), pp. 4338–4364. DOI: 10.1109/tpami.2020.3005434.
- [127] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Communications of the ACM* 60.6 (2017), pp. 84–90. DOI: 10.1145/3065386.
- [128] O. Ronneberger, P. Fischer, and T. Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

- [129] C. Olah. *Blog: Understanding LSTM Networks*. 2015. URL: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (visited on 03/29/2023).
- [130] D. Passos and P. Mishra. "A Tutorial on Automatic Hyperparameter Tuning of Deep Spectral Modelling for Regression and Classification Tasks". In: *Chemometrics and Intelligent Laboratory Systems* 223 (2022), p. 104520. ISSN: 0169-7439. DOI: 10.1016/j.chemolab.2022.104520.
- [131] F. Kaufmann, C. Glock, and T. Tschickardt. "ScaleBIM: Introducing a Scalable Modular Framework to Transfer Point Clouds into Semantically Rich Building Information Models". In: *Proceedings of the 2022 European Conference on Computing in Construction*. University of Turin, 2022. DOI: 10.35490/ec3.2022.194.
- [132] T. Hackel. "Large-scale Machine Learning for Point Cloud Processing". PhD thesis. ETH Zürich, Institute of Geodesy and Photogrammetry, 2018. DOI: 10.3929/ethz-b-000264691.
- [133] W. Zimmer, A. Rangesh, and M. Trivedi. "3D BAT: A Semi-Automatic, Web-based 3D Annotation Toolbox for Full-Surround, Multi-Modal Data Streams". In: *IEEE Intelligent Vehicles Symposium*. IEEE, 2019, pp. 1816–1821. DOI: 10.1109/ivs.2019.8814071.
- [134] Autodesk-Recap. *Youtube Channel*. 2021. URL: <http://https://www.youtube.com/user/autodeskreca/> (visited on 06/24/2021).
- [135] PointCab. *Punktwolken Software für alle Bedürfnisse*. 2023. URL: <https://pointcab-software.com> (visited on 01/23/2023).
- [136] J. Simon. *Blog: New - Label 3D Point Clouds with Amazon SageMaker Ground Truth*. Ed. by A. SageMaker. 2020. URL: <https://aws.amazon.com/de/blogs/aws/new-label-3d-point-clouds-with-amazon-sagemaker-ground-truth/> (visited on 01/23/2023).
- [137] basicAI. *AI Training Data Solutions for All Industries*. 2023. URL: <https://www.basic.ai/services> (visited on 01/23/2023).
- [138] scale. *Better Data*, 2023. URL: <https://scale.com/> (visited on 01/23/2023).
- [139] R. Richter. *Wie unsere Plattform funktioniert*. Ed. by P. C. Technology. 2023. URL: <https://www.pointcloudtechnology.com/de/#loesung> (visited on 01/23/2023).
- [140] R. Monica, J. Aleotti, M. Zillich, and M. Vincze. "Multi-label Point Cloud Annotation by Selection of Sparse Control Points". In: *International Conference on 3D Vision*. IEEE, 2017. DOI: 10.1109/3dv.2017.00042.
- [141] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. "MeshLab: An Open-Source Mesh Processing Tool". In: *Eurographics Italian Chapter Conference* (2008).
- [142] M. Veit and A. Capobianco. "Go Then Tag: A 3-D Point Cloud Annotation Technique". In: *IEEE Symposium on 3D User Interfaces*. IEEE, 2014. DOI: 10.1109/3dui.2014.6798886.

- [143] S. Kulkarni, M. Chandrashekararajah, and S. Raghunandan. “3D Annotation Tool Using LiDAR”. In: *IEEE Global Conference for Advancement in Technology*. Bangalore, India: IEEE, 2019, pp. 1–4. ISBN: 978-1-7281-3695-0. DOI: 10.1109/gcat47503.2019.8978301.
- [144] B. Wang, V. Wu, B. Wu, and K. Keutzer. “LATTE: Accelerating LiDAR Point Cloud Annotation via Sensor Fusion, One-Click Annotation, and Tracking”. In: *IEEE Intelligent Transportation Systems Conference*. IEEE, 2019. DOI: 10.1109/itsc.2019.8916980.
- [145] H. A. Arief, M. Arief, G. Zhang, Z. Liu, M. Bhat, U. G. Indahl, H. Tveite, and D. Zhao. “SAnE: Smart Annotation and Evaluation Tools for Point Cloud Data”. In: *IEEE Access* 8 (2020), pp. 131848–131858. DOI: 10.1109/access.2020.3009914.
- [146] D. Coffey, N. Malbraaten, T. B. Le, I. Borazjani, F. Sotiropoulos, A. G. Erdman, and D. F. Keefe. “Interactive Slice WIM: Navigating and Interrogating Volume Data Sets Using a Multisurface, Multitouch VR Interface”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.10 (2012), pp. 1614–1626. DOI: 10.1109/tvcg.2011.283.
- [147] N. O’Mahony, S. Campbell, A. Carvalho, L. Krpalkova, D. Riordan, and J. Walsh. “Point Cloud Annotation Methods for 3D Deep Learning”. In: *13th International Conference on Sensing Technology*. IEEE, 2019. DOI: 10.1109/icst46873.2019.9047730.
- [148] F. Wirth, J. Quehl, J. Ota, and C. Stiller. “PointAtMe: Efficient 3D Point Cloud Labeling in Virtual Reality”. In: *IEEE Intelligent Vehicles Symposium*. IEEE, 2019. DOI: 10.1109/ivs.2019.8814115.
- [149] S. Maneewongvatana and D. M. Mount. “Analysis of Approximate Nearest Neighbor Searching with Clustered Point Sets”. In: *ArXiv* (1999). DOI: 10.48550/ARXIV.CS/9901013.
- [150] J. Wilhelms and A. V. Gelder. “Octrees For Faster Isosurface Generation”. In: *Proceedings of the 1990 Workshop on Volume Visualization - VVS '90*. ACM Press, 1990. DOI: 10.1145/99307.99321.
- [151] R. B. Rusu, Z. C. Marton, N. Blodow, A. Holzbach, and M. Beetz. “Model-Based and Learned Semantic Object Labeling in 3D Point Cloud Maps of Kitchen Environments”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009. DOI: 10.1109/iros.2009.5354759.
- [152] H. Balta, J. Velagic, W. Bosschaerts, G. D. Cubber, and B. Siciliano. “Fast Statistical Outlier Removal Based Method for Large 3D Point Clouds of Outdoor Environments”. In: *IFAC-PapersOnLine* 51.22 (2018), pp. 348–353. DOI: 10.1016/j.ifacol.2018.11.566.
- [153] J. Zhu, J. Gehring, R. Huang, B. Borgmann, Z. Sun, L. Hoegner, M. Hebel, Y. Xu, and U. Stilla. “TUM-MLS-2016: An Annotated Mobile LiDAR Dataset of the TUM City Campus for Semantic Point Cloud Interpretation in Urban Areas”. In: *Remote Sensing* 12.11 (2020), p. 1875. DOI: 10.3390/rs12111875.

- [154] CloudCompare. *3D Point Cloud and Mesh Processing Software Open-Source Project*. Version 2.12. 2021. URL: <http://www.cloudcompare.org/> (visited on 06/24/2021).
- [155] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. “Dynamic Graph CNN for Learning on Point Clouds”. In: *ACM Transactions on Graphics* 38.5 (2019), pp. 1–12. DOI: 10.1145/3326362.
- [156] K. Wada. *Labelme: Image Polygonal Annotation with Python*. 2016. DOI: 10.5281/zenodo.5711226. URL: <https://github.com/wkentaro/labelme> (visited on 07/15/2021).
- [157] E. Gaur, V. Saxena, and S. K. Singh. “Video Annotation Tools: A Review”. In: *IEEE International Conference on Advances in Computing, Communication Control and Networking*. IEEE, 2018. DOI: 10.1109/icacccn.2018.8748669.
- [158] X. Roynard, J.-E. Deschaud, and F. Goulette. “Paris-Lille-3D: A Large and High-Quality Ground-Truth Urban Point Cloud Dataset for Automatic Segmentation and Classification”. In: *The International Journal of Robotics Research* 37.6 (2018), pp. 545–557. DOI: 10.1177/0278364918767506.
- [159] W. Tan, N. Qin, L. Ma, Y. Li, J. Du, G. Cai, K. Yang, and J. Li. “Toronto-3D: A Large-scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways”. In: *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2020. DOI: 10.1109/cvprw50498.2020.00109.
- [160] C. Wang, Y. Dai, N. Elsheimy, C. Wen, G. Retscher, Z. Kang, and A. Lingua. “ISPRS Benchmark on multisensory indoor mapping and positioning”. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences V-5-2020* (2020), pp. 117–123. DOI: 10.5194/isprs-annals-v-5-2020-117-2020.
- [161] G. Poier, M. Seidl, M. Zeppelzauer, C. Reinbacher, M. Schaich, G. Bellandi, A. Marretta, and H. Bischof. “PetroSurf3D – A Dataset for high-resolution 3D Surface Segmentation”. In: *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*. ACM, 2017. DOI: 10.1145/3095713.3095719.
- [162] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang. “Deep Learning on Point Clouds and Its Application: A Survey”. In: *Sensors* 19.19 (2019), p. 4188. DOI: 10.3390/s19194188.
- [163] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao. “Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2020), pp. 6063–6081. ISSN: 1524-9050. DOI: 10.1109/tits.2021.3076844.
- [164] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. *ShapeNet: An Information-Rich 3D Model Repository*. Tech. rep. Stanford University, Princeton University, Toyota Technological Institute at Chicago, 2015. arXiv: 1512.03012.

- [165] J. Iqbal, R. Xu, S. Sun, and C. Li. "Simulation of an Autonomous Mobile Robot for LiDAR-Based In-Field Phenotyping and Navigation". In: *Robotics* 9.2 (2020), p. 46. DOI: 10.3390/robotics9020046.
- [166] L. Winiwarter, A. M. E. Pena, H. Weiser, K. Anders, J. M. Sánchez, M. Searle, and B. Höfle. "Virtual Laser Scanning with HELIOS++: A Novel Take on Ray Tracing-Based Simulation of Topographic Full-Waveform 3D Laser Scanning". In: *Remote Sensing of Environment* 269 (2021), p. 112772. DOI: 10.1016/j.rse.2021.112772.
- [167] V. Stojanovic, M. Trapp, J. Döllner, and R. Richter. "Classification of Indoor Point Clouds Using Multiviews". In: *The 24th International Conference on 3D Web Technology*. ACM, 2019. DOI: 10.1145/3329714.3338129.
- [168] J. Gehrung, M. Hebel, M. Arens, and U. Stilla. "An Approach to Extract Moving Objects from MLS Data Using a Volumetric Background Representation". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* IV-1/W1 (2017), pp. 107–114. DOI: 10.5194/isprs-annals-iv-1-w1-107-2017.
- [169] E. Grilli, F. Menna, and F. Remondino. "A Review of Point Clouds Segmentation and Classification Algorithms". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2/W3 (2017), pp. 339–344. DOI: 10.5194/isprs-archives-xlii-2-w3-339-2017.
- [170] P. Klinger. "Übersicht und Bewertung von aktuellen Segmentierungs- und Klassifizierungsmethoden von Punktwolken". Bachelorthesis. HafenCity University Hamburg, 2019.
- [171] P. Babahajiani, L. Fan, J.-K. Kämäräinen, and M. Gabbouj. "Comprehensive Automated 3D Urban Environment Modelling Using Terrestrial Laser Scanning Point Cloud". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016. DOI: 10.1109/cvprw.2016.87.
- [172] O. Hassaan, A. Shamaail, Z. Butt, and M. Taj. "Point Cloud Segmentation using Hierarchical Tree for Architectural Models". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 1582–1586. DOI: 10.1109/icassp.2019.8682708.
- [173] E. Castillo, J. Liang, and H. Zhao. "Point Cloud Segmentation and Denoising via Constrained Nonlinear Least Squares Normal Estimates". In: *Mathematics and Visualization*. Springer Berlin Heidelberg, 2012, pp. 283–299. DOI: 10.1007/978-3-642-34141-0_13.
- [174] P. J. Besl and R. C. Jain. "Segmentation through Variable-Order Surface Fitting". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.2 (1988), pp. 167–192. DOI: 10.1109/34.3881.
- [175] F. Pauling, M. Bosse, and R. Zlot. "Automatic Segmentation of 3D Laser Point Clouds by Ellipsoidal Region Growing". In: *Australasian Conference on Robotics and Automation*. Sydney, Australia, 2009.

- [176] J. Strom, A. Richardson, and E. Olson. "Graph-Based Segmentation for Colored 3D Laser Point Clouds". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2131-2136. IEEE, 2010. DOI: 10.1109/iros.2010.5650459.
- [177] A. Golovinskiy and T. Funkhouser. "Min-Cut Based Segmentation of Point Clouds". In: *12th International Conference on Computer Vision Workshops*. IEEE, 2009, pp. 39–46. DOI: 10.1109/iccvw.2009.5457721.
- [178] F. Moosmann, O. Pink, and C. Stiller. "Segmentation of 3D Lidar Data in non-flat Urban Environments using a Local Convexity Criterion". In: *IEEE Intelligent Vehicles Symposium*. IEEE, 2009, pp. 215–220. DOI: 10.1109/ivs.2009.5164280.
- [179] A. Aijazi, P. Checchin, and L. Trassoudaine. "Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach with Evaluation". In: *Remote Sensing* 5.4 (2013), pp. 1624–1650. DOI: 10.3390/rs5041624.
- [180] P. V. C. Hough. "Method and Means for Recognizing Complex Patterns". US Patent 3069654. 1962.
- [181] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter. "The 3d Hough Transform for Plane Detection in Point Clouds: A Review and a New Accumulator Design". In: *3D Research* 2.2 (2011). ISSN: 2092-6731. DOI: 10.1007/3dres.02(2011)3.
- [182] W. Shi, W. Ahmed, N. Li, W. Fan, H. Xiang, and M. Wang. "Semantic Geometric Modelling of Unstructured Indoor Point Cloud". In: *ISPRS International Journal of Geo-Information* 8.1 (2018), p. 9. DOI: 10.3390/ijgi8010009.
- [183] M. Weinmann, B. Jutzi, and C. Mallet. "Feature Relevance Assessment for the Semantic Interpretation of 3D Point Cloud Data". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II-5/W2 (2013), pp. 313–318. DOI: 10.5194/isprsannals-ii-5-w2-313-2013.
- [184] M. Weinmann, B. Jutzi, and C. Mallet. "Semantic 3D Scene Interpretation: A Framework Combining Optimal Neighborhood Size Selection with Relevant Features". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II-3 (2014), pp. 181–188. DOI: 10.5194/isprsannals-ii-3-181-2014.
- [185] M. Weinmann, A. Schmidt, C. Mallet, S. Hinz, F. Rottensteiner, and B. Jutzi. "Contextual Classification of Point Cloud Data by Exploiting Individual 3d Neighbourhoods". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* II-3/W4 (2015), pp. 271–278. DOI: 10.5194/isprsannals-ii-3-w4-271-2015.
- [186] H. Thomas, F. Goulette, J.-E. Deschaud, B. Marcotegui, and Y. LeGall. "Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods". In: *IEEE International Conference on 3D Vision*. IEEE, 2018. DOI: 10.1109/3dv.2018.00052.
- [187] Z. Lari and A. Habib. "Alternative Methodologies for the Estimation of Local Point Density Index: Moving Towards Adaptive Lidar Data Processing". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXIX-B3 (2012), pp. 127–132. DOI: 10.5194/isprsarchives-xxxix-b3-127-2012.

- [188] F. Poux and R. Billen. “Voxel-based 3D Point Cloud Semantic Segmentation: Un-supervised Geometric and Relationship Featuring vs. Deep Learning Methods”. In: *ISPRS International Journal of Geo-Information* 8.5 (2019), p. 213. DOI: 10.3390/ijgi8050213.
- [189] F. Poux, C. Mattes, and L. Kobbelt. “Unsupervised Segmentation of Indoor 3D Point Cloud: Application to Object-Based Classification”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIV-4/W1-2020* (2020), pp. 111–118. DOI: 10.5194/isprs-archives-xliv-4-w1-2020-111-2020.
- [190] E. Grilli, D. Dinunno, G. Petrucci, and F. Remondino. “From 2D to 3D Supervised Segmentation and Classification for Cultural Heritage Applications”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2* (2018), pp. 399–406. DOI: 10.5194/isprs-archives-xlii-2-399-2018.
- [191] S. Teruggi, E. Grilli, M. Russo, F. Fassi, and F. Remondino. “A Hierarchical Machine Learning Approach for Multi-Level and Multi-Resolution 3D Point Cloud Classification”. In: *Remote Sensing* 12.16 (2020), p. 2598. DOI: 10.3390/rs12162598.
- [192] E. Grilli, F. Poux, and F. Remondino. “Unsupervised Object-Based Clustering in Support of Supervised Point-Based 3d Point Cloud Classification”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B2-2021* (2021), pp. 471–478. DOI: 10.5194/isprs-archives-xliii-b2-2021-471-2021.
- [193] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin. “Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR Segmentation”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2020. DOI: 10.1109/cvpr46437.2021.00981.
- [194] H. Radi and W. Ali. “VolMap: A Real-time Model for Semantic Segmentation of a LiDAR surrounding view”. In: *ICML (Thirty-sixth International Conference on Machine Learning) Workshop on AI for Autonomous Driving*. arXiv, 2019.
- [195] R. A. Rosu, P. Schütt, J. Quenzel, and S. Behnke. “LatticeNet: Fast Point Cloud Segmentation Using Permutohedral Lattices”. In: *Proceedings of Robotics: Science and Systems*. Robotics: Science and Systems Foundation, 2020. DOI: 10.15607/rss.2020.xvi.006.
- [196] H.-Y. Meng, L. Gao, Y.-K. Lai, and D. Manocha. “VV-Net: Voxel VAE Net With Group Convolutions for Point Cloud Segmentation”. In: *International Conference on Computer Vision*. IEEE, 2019. DOI: 10.1109/iccv.2019.00859.
- [197] A. Krizhevsky. “One weird trick for parallelizing convolutional neural networks”. In: *arXiv* (2014). DOI: 10.48550/ARXIV.1404.5997. arXiv: 1404.5997.
- [198] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. “You Only Look Once: Unified, Real-Time Object Detection”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 779–788. DOI: 10.1109/cvpr.2016.91.

- [199] J. Redmon and A. Farhadi. "YOLOv3: An Incremental Improvement". In: *arXiv* 2.4 (2018). DOI: 10.48550/ARXIV.1804.02767.
- [200] E. Camuffo, D. Mari, and S. Milani. "Recent Advancements in Learning Algorithms for Point Clouds: An Updated Overview". In: *Sensors* 22.4 (2022), p. 1357. DOI: 10.3390/s22041357.
- [201] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde. "Simultaneous Perception and Path Generation Using Fully Convolutional Neural Networks". In: *20th International Conference on Intelligent Transportation Systems*. IEEE, 2017. DOI: 10.1109/itsc.2017.8317618.
- [202] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde. "Fast LIDAR-based Road Detection Using Fully Convolutional Neural Networks". In: *Intelligent Vehicles Symposium IV*. IEEE, 2017. DOI: 10.1109/ivs.2017.7995848.
- [203] L. Gigli, B. R. Kiran, T. Paul, A. Serna, N. Vemuri, B. Marcotegui, and S. Velasco-Forero. "Road Segmentation on Low Resolution Lidar Point Clouds for Autonomous Vehicles". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-2-2020* (2020), pp. 335–342. DOI: 10.5194/isprs-annals-v-2-2020-335-2020.
- [204] J. Fritsch, T. Kuehnl, and A. Geiger. "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms". In: *International Conference on Intelligent Transportation Systems*. IEEE, 2013. DOI: 10.1109/itsc.2013.6728473.
- [205] H. Chu, W.-C. Ma, K. Kundu, R. Urtasun, and S. Fidler. "SurfConv: Bridging 3D and 2D Convolution for RGBD Images". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2018. DOI: 10.1109/cvpr.2018.00317.
- [206] B. Yang, W. Luo, and R. Urtasun. "PIXOR: Real-Time 3D Object Detection From Point Clouds". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2018, pp. 7652–7660. DOI: 10.1109/cvpr.2018.00798.
- [207] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2019. DOI: 10.1109/iros40897.2019.8967762.
- [208] P. Biasutti, V. Lepetit, J.-F. Aujol, M. Bredif, and A. Bugeau. "LU-Net: An Efficient Network for 3D LiDAR Point Cloud Semantic Segmentation Based on End-to-End-Learned 3D Features and U-Net". In: *International Conference on Computer Vision Workshop*. IEEE, 2019. DOI: 10.1109/iccvw.2019.00123.
- [209] T. Cortinhal, G. Tzelepis, and E. E. Aksoy. "SalsaNext: Fast, Uncertainty-aware Semantic Segmentation of LiDAR Point Clouds for Autonomous Driving". In: *Advances in Visual Computing*. Springer International Publishing, 2020, pp. 207–222. DOI: 10.1007/978-3-030-64559-5_16.

- [210] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. "Squeezenet: Alexnet-Level Accuracy with 50x Fewer Parameters and <1MB Model Size". In: *ArXiv abs/1602.07360* (2016).
- [211] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2016. DOI: 10.1109/cvpr.2016.90.
- [212] A. Boulch, B. Le Saux, and N. Audebert. "Unstructured Point Cloud Semantic Labeling Using Deep Segmentation Networks". In: *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2017. DOI: 10.2312/3DOR.20171047.
- [213] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou. "Tangent Convolutions for Dense Prediction in 3D". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2018. DOI: 10.1109/cvpr.2018.00409.
- [214] R. Kaijaluoto, A. Kukko, A. E. Issaoui, J. Hyppä, and H. Kaartinen. "Semantic Segmentation of Point Cloud Data Using Raw Laser Scanner Measurements and Deep Neural Networks". In: *ISPRS Open Journal of Photogrammetry and Remote Sensing* 3 (2022), p. 100011. DOI: 10.1016/j.ojphoto.2021.100011.
- [215] A. Reiterer, K. Wäschle, D. Störk, A. Leydecker, and N. Gitzen. "Fully Automated Segmentation of 2D and 3D Mobile Mapping Data for Reliable Modeling of Surface Structures Using Deep Learning". In: *Remote Sensing* 12.16 (2020), p. 2530. DOI: 10.3390/rs12162530.
- [216] M. Voelsen, M. Teimouri, F. Rottensteiner, and C. Heipke. "Investigating 2D and 3D Convolutions for Multitemporal Land Cover Classification Using Remote Sensing Images". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-3-2022* (2022), pp. 271–279. DOI: 10.5194/isprs-annals-v-3-2022-271-2022.
- [217] V. Jampani, M. Kiefel, and P. V. Gehler. "Learning Sparse High Dimensional Filters: Image Filtering, Dense CRFs and Bilateral Neural Networks". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2016. DOI: 10.1109/cvpr.2016.482.
- [218] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz. "SPLAT-Net: Sparse Lattice Networks for Point Cloud Processing". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2018. DOI: 10.1109/cvpr.2018.00268.
- [219] J. Huang and S. You. "Point Cloud Labeling using 3D Convolutional Neural Network". In: *International Conference on Pattern Recognition*. IEEE, 2016. DOI: 10.1109/icpr.2016.7900038.
- [220] H. Moravec and A. Elfes. "High Resolution Maps from Wide Angle Sonar". In: *Proceedings IEEE International Conference on Robotics and Automation*. IEEE, 1985. DOI: 10.1109/robot.1985.1087316.

- [221] C. Köhler, M. Donner, and R. Donner. “Semantische Klassifizierung von 3D-Punktwolken”. In: *19. Geokinematischer Tag des Institutes für Markscheidewesen und Geodäsie*. Freiberg: Wagner Digitaldruck und Medien GmbH, 2018, pp. 117–130. ISBN: 9783938390214. DOI: 10.1007/978-3-662-46251-5_10.
- [222] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *3rd International Conference on Learning Representations*. San Diego, USA: Computational and Biological Learning Society, 2015, pp. 1–14.
- [223] D. Zeng Wang and I. Posner. “Voting for Voting in Online Point Cloud Object Detection”. In: *Robotics: Science and Systems XI*. Robotics: Science and Systems Foundation, 2015. DOI: 10.15607/rss.2015.xi.035.
- [224] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. “Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks”. In: *International Conference on Robotics and Automation*. IEEE, 2017. DOI: 10.1109/icra.2017.7989161.
- [225] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. “Semantic Scene Completion from a Single Depth Image”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017, pp. 190–198. DOI: 10.1109/cvpr.2017.28.
- [226] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017. DOI: 10.1109/cvpr.2017.261.
- [227] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. “ScanComplete: Large-Scale Scene Completion and Semantic Segmentation for 3D Scans”. In: *Conference on Computer Vision and Pattern Recognition*. Vol. 1. IEEE, 2018, p. 2. DOI: 10.1109/cvpr.2018.00481.
- [228] L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese. “SEGCloud: Semantic Segmentation of 3D Point Clouds”. In: *International Conference on 3D Vision*. IEEE, 2017. DOI: 10.1109/3dv.2017.00067.
- [229] G. Riegler, A. O. Ulusoy, and A. Geiger. “OctNet: Learning Deep 3D Representations at High Resolutions”. In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017. DOI: 10.1109/cvpr.2017.701.
- [230] D. Meagher. “Geometric Modeling Using Octree Encoding”. In: *Computer Graphics and Image Processing* 19.1 (1982), p. 85. DOI: 10.1016/0146-664x(82)90128-9.
- [231] J. L. Bentley. “Multidimensional Binary Search Trees Used for Associative Searching”. In: *Communications of the ACM* 18.9 (1975), pp. 509–517. DOI: 10.1145/361002.361007.
- [232] R. Klokov and V. Lempitsky. “Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models”. In: *International Conference on Computer Vision*. IEEE, 2017. DOI: 10.1109/iccv.2017.99.

- [233] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5099–5108.
- [234] F. Engelmann, T. Kontogiannia, A. Hermans, and B. Leibe. "Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds". In: *International Conference on Computer Vision Workshops*. IEEE, 2017, pp. 716–724. DOI: 10.1109/iccvw.2017.90. eprint: 1802.01500.
- [235] J. Li, B. M. Chen, and G. H. Lee. "SO-Net: Self-Organizing Network for Point Cloud Analysis". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2018. DOI: 10.1109/cvpr.2018.00979.
- [236] Z. Zhao, M. Liu, and K. Ramani. "DAR-Net: Dynamic Aggregation Network for Semantic Scene Segmentation". In: *arXiv* (2019). eprint: 1907.12022.
- [237] F. Engelmann, T. Kontogianni, J. Schult, and B. Leibe. "Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2019, pp. 395–409. DOI: 10.1007/978-3-030-11015-4_29.
- [238] Y. Rao, M. Zhang, Z. Cheng, J. Xue, J. Pu, and Z. Wang. "Semantic Point Cloud Segmentation Using Fast Deep Neural Network and DCRF". In: *Sensors* 21.8 (2021), p. 2731. DOI: 10.3390/s21082731.
- [239] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu. "PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation". In: *arXiv* (2018). DOI: <https://doi.org/10.48550/arXiv.1807.00652>.
- [240] L.-Z. Chen, X.-Y. Li, D.-P. Fan, K. Wang, S.-P. Lu, and M.-M. Cheng. "LSANet: Feature Learning on Point Sets by Local Spatial Aware Layer". In: *arXiv* (2019). DOI: 10.48550/ARXIV.1905.05442.
- [241] C. Chen, L. Z. Fragonara, and A. Tsourdos. "Go Wider: An Efficient Neural Network for Point Cloud Analysis via Group Convolutions". In: *Applied Sciences* 10.7 (2020), p. 2391. DOI: 10.3390/app10072391.
- [242] Q. Huang, W. Wang, and U. Neumann. "Recurrent Slice Networks for 3D Segmentation of Point Clouds". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2018. DOI: 10.1109/cvpr.2018.00278.
- [243] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. "PointCNN: Convolution On X-Transformed Points". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates Inc., 2018, pp. 828–838.
- [244] Z. Zhang, B.-S. Hua, and S.-K. Yeung. "ShellNet: Efficient Point Cloud Convolutional Neural Networks Using Concentric Shells Statistics". In: *International Conference on Computer Vision*. IEEE, 2019. DOI: 10.1109/iccv.2019.00169.

- [245] W. Wu, Z. Qi, and L. Fuxin. "PointConv: Deep Convolutional Networks on 3D Point Clouds". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2019. DOI: 10.1109/cvpr.2019.00985.
- [246] A. Komarichev, Z. Zhong, and J. Hua. "A-CNN: Annularly Convolutional Neural Networks on Point Clouds". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2019. DOI: 10.1109/cvpr.2019.00760.
- [247] F. Engelmann, T. Kontogianni, and B. Leibe. "Dilated Point Convolutions: On the Receptive Field Size of Point Convolutions on 3D Point Clouds". In: *IEEE International Conference on Robotics and Automation*. IEEE, 2020. DOI: 10.1109/icra40945.2020.9197503.
- [248] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. "Graph Neural Networks: A Review of Methods and Applications". In: *AI Open 1* (2020), pp. 57–81. DOI: 10.1016/j.aiopen.2021.01.001.
- [249] S. A. Khan, Y. Shi, M. Shahzad, and X. X. Zhu. "FGCN: Deep Feature-based Graph Convolutional Network for Semantic Segmentation of Urban 3D Point Clouds". In: *Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2020. DOI: 10.1109/cvprw50498.2020.00107.
- [250] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia. "Hierarchical Point-Edge Interaction Network for Point Cloud Semantic Segmentation". In: *International Conference on Computer Vision*. Seoul, Korea (South): IEEE, 2019, pp. 10432–10440. DOI: 10.1109/iccv.2019.01053.
- [251] C. Morbidoni, R. Pierdicca, R. Quattrini, and E. Frontoni. "Graph CNN with Radius Distance for Semantic Segmentation of Historical Buildings TLS Point Clouds". In: *ISPRS The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIV-4/W1-2020* (2020), pp. 95–102. DOI: 10.5194/isprs-archives-xliv-4-w1-2020-95-2020.
- [252] E. Widyaningrum, Q. Bai, M. K. Fajari, and R. C. Lindenbergh. "Airborne Laser Scanning Point Cloud Classification Using the DGCNN Deep Learning Method". In: *Remote Sensing 13.5* (2021), p. 859. DOI: 10.3390/rs13050859.
- [253] 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". In: *International Journal of Applied Earth Observation and Geoinformation 105* (2021), p. 102626. DOI: 10.1016/j.jag.2021.102626.
- [254] L. Landrieu and M. Simonovsky. "Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2018. DOI: 10.1109/cvpr.2018.00479.

- [255] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. "Tensorflow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". In: *arXiv* (2016). DOI: 10.48550/ARXIV.1603.04467.
- [256] F. J. J. Joseph, S. Nonsiri, and A. Monsakul. "Keras and TensorFlow: A Hands-On Experience". In: *Advanced Deep Learning for Engineers and Scientists*. Springer International Publishing, 2021, pp. 85–111. DOI: 10.1007/978-3-030-66519-7_4.
- [257] HexagonMetrology. *Product brochure, Leica T-Scan TS 50-A*. 2021. URL: https://w3.leica-geosystems.com/downloads123/m1/metrology/t-scan/brochures/leica%20t-scan%20brochure_en.pdf (visited on 06/24/2021).
- [258] J. M. Johnson and T. M. Khoshgoftaar. "Survey on Deep Learning with Class Imbalance". In: *Journal of Big Data* 6.1 (2019). DOI: 10.1186/s40537-019-0192-5.
- [259] M. R. Rezaei-Dastjerdehei, A. Mijani, and E. Fatemizadeh. "Addressing Imbalance in Multi-Label Classification Using Weighted Cross Entropy Loss Function". In: *27th National and 5th International Iranian Conference on Biomedical Engineering*. IEEE, 2020. DOI: 10.1109/icbme51989.2020.9319440.
- [260] Z. Jin, Y. Lei, N. Akhtar, H. Li, and M. Hayat. "Deformation and Correspondence Aware Unsupervised Synthetic-to-Real Scene Flow Estimation for Point Clouds". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2022. DOI: 10.1109/cvpr52688.2022.00709.
- [261] 3D-Systemes. *Geomagic Wrap*. 2023. URL: <https://de.3dsystems.com/software/geomagic-wrap> (visited on 04/05/2023).
- [262] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. "Introduction to WordNet: An On-line Lexical Database". In: *International Journal of Lexicography* 3.4 (1990), pp. 235–244. DOI: 10.1093/ijl/3.4.235.
- [263] L. Gao, J. Song, F. Nie, F. Zou, N. Sebe, and H. T. Shen. "Graph-without-cut: An Ideal Graph Learning for Image Segmentation." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1. Association for the Advancement of Artificial Intelligence, 2016, p. 6. DOI: 10.1609/aaai.v30i1.10177.
- [264] E. Barnefske and H. Sternberg. "Generation of Training Data for 3D Point Cloud Classification by CNN". In: *Proceedings of 80th FIG Working Week 2019*. Hanoi, Vietnam, 2019.

- [265] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu. "Shape2Motion: Joint Analysis of Motion Parts and Attributes from 3D Shapes". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2019, pp. 8868–8876. DOI: 10.1109/cvpr.2019.00908.
- [266] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. "Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition". In: *Neural networks*. Vol. 32. Elsevier BV, 2012, pp. 323–332. DOI: 10.1016/j.neunet.2012.02.016.
- [267] G. Joos. "Zur Qualität von objektstrukturierten Geodaten". PhD thesis. Universität der Bundeswehr München, 2000.
- [268] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. "3D Semantic Parsing of Large-Scale Indoor Spaces". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 1534–1543. DOI: 10.1109/cvpr.2016.170.
- [269] M. A. Uy, Q.-H. Pham, B.-S. Hua, D. T. Nguyen, and S.-K. Yeung. "Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data". In: *International Conference on Computer Vision*. IEEE, 2019. DOI: 10.1109/iccv.2019.00167.
- [270] T. Yu and H. Zhu. "Hyper-Parameter Optimization: A Review of Algorithms and Applications". In: *arXiv arXiv:2003.05689 (2020)*. arXiv: 2003.05689v1 [cs.LG].
- [271] D. Griffiths and J. Boehm. "Weighted Point Cloud Augmentation for Neural Network Training Data Class-Imbalance". In: *ISPRS The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W13 (2019)*, pp. 981–987. DOI: 10.5194/isprs-archives-XLII-2-W13-981-2019.
- [272] J. Morel, A. Bac, and T. Kanai. "Segmentation of Unbalanced and In-Homogeneous Point Clouds and Its Application to 3D Scanned Trees". In: *The Visual Computer* 36.10-12 (2020), pp. 2419–2431. DOI: 10.1007/s00371-020-01966-7.
- [273] V. Kasireddy and B. Akinci. "Assessing the Impact of 3D Point Neighborhood Size Selection on Unsupervised Spall Classification with 3D Bridge Point Clouds". In: *Advanced Engineering Informatics* 52 (2022), p. 101624. DOI: 10.1016/j.aei.2022.101624.
- [274] Y. Liu, B. Fan, G. Meng, J. Lu, S. Xiang, and C. Pan. "DensePoint: Learning Densely Contextual Representation for Efficient Point Cloud Processing". In: *International Conference on Computer Vision*. IEEE, 2019. DOI: 10.1109/iccv.2019.00534.
- [275] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung. "SceneNN: A Scene Meshes Dataset with aNNotations". In: *Fourth International Conference on 3D Vision*. IEEE, 2016, pp. 92–101. DOI: 10.1109/3dv.2016.18.
- [276] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia. "PointWeb: Enhancing Local Neighborhood Features for Point Cloud Processing". In: *Conference on Computer Vision and Pattern Recognition*. IEEE, 2019. DOI: 10.1109/cvpr.2019.00571.

- [277] S. Ando and C. Y. Huang. “Deep Over-sampling Framework for Classifying Imbalanced Data”. In: *European Conference, ECML PKDD*. Springer International Publishing, Cham, 2017, pp. 770–785. DOI: 10.1007/978-3-319-71249-9_46.
- [278] M. Joseph-Rivlin, A. Zvirin, and R. Kimmel. “Momenet: Flavor the Moments in Learning to Classify Shapes”. In: *International Conference on Computer Vision Workshop*. IEEE, 2019. DOI: 10.1109/iccvw.2019.00503.

A Peer-reviewed publications

A.1 Klassifizierung von fehlerhaft gemessenen Punkten in 3D-Punktwolken mit ConvNet

Reference:

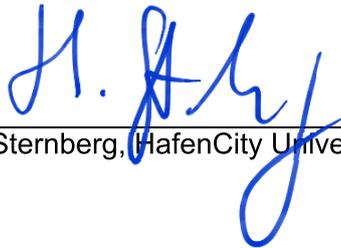
Barnefske, E. & Sternberg, H. (2020): Klassifizierung von fehlerhaft gemessenen Punkten in 3D-Punktwolken mit ConvNet. In Wunderlich, T. (Ed.), Ingenieurvermessung 20. Beiträge zum 19. Internationalen Ingenieurvermessungskurs München, 2020, Herbert Wichmann Verlag, 2020, 127-139.

Contribution of Co-Authors:

Table 6: Contribution to Paper No. 1

Involved in	Estimated contribution
Ideas and conceptual design	90%
Computation and results	100%
Analysis and interpretation	95%
Manuscript, figures and tables	100%
Total:	96%

I hereby confirm the correctness of the declaration of the contribution of Eike Barnefske for Paper No. 1 in Table 6:



Prof. Dr.-Ing. Harald Sternberg, HafenCity Universität Hamburg

Klassifizierung von fehlerhaft gemessenen Punkten in 3D-Punktwolken mit ConvNet

Eike BARNEFSKE und Harald STERNBERG

1 Einleitung

Punktwolken dienen als einfaches Modell oder als Grundlage von Planungen, geometrischen Analysen und komplexen Modellierungen. Häufig sind im ersten Schritt Teilpunktwolken für diese Aufgaben zu erstellen. Eine Filterung nach fehlerhaft gemessenen Punkten oder ein modellbasiertes Reduzieren der Punktwolkendichte sind häufig die ersten Klassifizierungen, die auf eine Punktwolke angewendet werden. Hierfür werden vorrangig händische oder modellbasierte Verfahren genutzt. Alternativ zu modell- bzw. wissensbasierten Klassifizierungsverfahren werden datenbasierte Klassifizierungen entwickelt, um die Auswertungszeit zu reduzieren, die Klassifizierungsqualität bei unterschiedlichen Aufnahmesystemen und Szenen zu steigern, sowie um die Klassifizierung zu automatisieren. Hierfür finden u. a. Convolutional Neuronale Netzwerke (ConvNet) Anwendung, die das Wissen aus den klassifizierten Punktwolken (Trainingsdaten) lernen und dieses in der Anwendungsphase nutzen, um unbekannte Punkte zu klassifizieren. Die Klassifizierungsleistung der ConvNet wird folglich durch die Netzwerkarchitektur und die Trainingsdaten (Wissen des Algorithmus) beeinflusst.

Arbeiten von QI ET AL. (2017A), HACKEL ET AL. (2017) u. a. zur semantischen Klassifizierung mit ConvNet richten sich an folgendes Ziel: Es sollen die Punkte identifiziert werden, welche bestimmte Objekte in der Aufnahmeszene (z. B. Bäume oder Straßen) beschreiben. Hierbei werden aber fehlerhaft gemessene Punkte, die i. d. R. nur wenige Prozent der gesamten Punktwolke einnehmen, nicht betrachtet und einfach einer Objektklasse zugeordnet. Diese fehlerhaften Punkte treten z. B. in der Form von Mixed Pixel, Mehrwegeeffekten, einem großen Oberflächenrauschen oder Phantompunkten auf.

Der Einfluss der fehlerhaft gemessenen Punkte auf die semantische Klassifizierung und eine Klassifizierung nach fehlerhaften gemessenen Punkten ist Gegenstand der Untersuchungen dieser Arbeit. Hierfür werden händisch klassifizierte Punktwolken in unterschiedlichen Klassenkombinationen für das Training und die Evaluierung der Klassifizierungsleistung verwendet.

Es werden die Unterschiede und Ähnlichkeiten von modell- und datenbasierten Klassifikationsverfahren vorgestellt (Abschnitt 2.1). Die Funktionsweise von ConvNet (Abschnitt 2.2) und verschiedene ConvNet-Modelle für strukturierte und unstrukturierte Punktwolken werden erläutert (Abschnitte 2.3 und 2.4). Der Einfluss bei vier unterschiedlichen Klassenkombinationen mit und ohne fehlerhaften Punkten auf die Klassifizierungsleistung wird am Beispiel der Netzwerkarchitektur von *PointNet* untersucht (Abschnitte 3.1 bis 3.3). Aufbauend auf den Beobachtungen der Untersuchungen werden Strategien für das Training von ConvNet zur Klassifizierung von fehlerhaft gemessenen Punkten vorgestellt. Zudem werden Ideen vorgestellt, um den Einfluss von fehlerhaft gemessenen Punkten bei Klassifizierungen zu minimieren (Abschnitt 3.4).

2 Klassifizierung von Punktwolken

Punktwolken stellen geometrisch einen erfassten Raum dar, lassen aber ohne weiteres Wissen eine semantische Trennung von einzelnen Objekten in der erfassten Szene nicht zu. Diese semantische Zerlegung der Punktwolke in Unterpunktwolken ist ein wichtiger Schritt des Auswertungsprozesses von Punktwolken, damit zum einen fehlerhaft gemessene Punkte aus der Punktwolke entfernt werden und zum anderen Objekte in einer Szene semantisch unterschieden werden können. Eine semantische Unterscheidung ist wichtig, da nicht alle erfassten Objekte für eine Fragestellung von Interesse sind. Für die Erstellung von Stadtmodellen sind z. B. Bauwerke von Interesse, Fahrzeuge hingegen werden hier als störende Objekte deklariert. Soll die Punktwolke für die Analyse des Verkehrsraums, z. B. bei der Verkehrsplanung oder der Navigation, verwendet werden, sind Punkte, die Objekte auf den Verkehrswegen (Autos, Fußgänger und Radfahrer) erfassen, von vornehmlichem Interesse.

Die Trennung von Punkten, die Objekte beschreiben und Punkten, die aufgrund von Messfehlern entstanden sind, wird i. d. R. mit Filtern durchgeführt. Diese Filter nutzen allgemeine oder sensorspezifische Modelle zur Unterscheidung, ob ein Punkt zu einem Objekt gehört oder aufgrund eines Messfehlers entstanden ist. Zudem werden die Filter zum Homogenisieren der Punktwolkendichte und zur Auswahl von Punktwolkenabschnitten eingesetzt. Filter haben den Nachteil, dass viel Wissen über die Punktwolke vorhanden sein muss und dieses für jeden möglichen Fall angewendet werden muss.

In einem nachfolgenden Schritt kommen i. d. R. andere Modelle zur automatischen semantischen Trennung der Punktwolken nach Objektklassen zum Einsatz. Beispielhaft werden diese modellbasierten Verfahren vorgestellt. Im Gegensatz zu den modellbasierten Klassifizierungsverfahren werden vermehrt datenbasierte Klassifizierungsverfahren entwickelt, die die Trennung der Punktwolken nicht aufgrund von vorgegebenem Wissen, sondern von erlerntem Wissen für die semantische Trennung der Punktwolken durchführen. Den populärsten Ansatz stellen zurzeit ConvNet dar, dessen Einsatz an einigen Beispielen vorgestellt wird.

2.1 Modellbasierte Segmentierung und Klassifizierung

Die Klassifizierung von Punktwolken basiert auf den vier zentralen Arbeitsschritten, und zwar (1) der Detektion von Merkmalen, (2) dem Sortieren der Punkte nach diesen Merkmalen, (3) dem Festlegen von Grenzen, die die Gruppen (Segmente) mit ähnlichen Merkmalen voneinander unterscheiden, und (4) dem Zuweisen eines Klassennamens an alle Segmente mit gleichen bzw. ähnlichen Merkmalen. Die Arbeitsschritte 1 bis 3 werden als Segmentierung bezeichnet, auf die eine Klassifikation folgen kann. Bei modellbasierten Verfahren können diese Arbeitsschritte i. d. R. eindeutig von dem der Klassifikation unterschieden werden. Dies ist bei datenbasierten Verfahren zunehmend nicht mehr möglich, da die Generierung der Segmente aufgrund von Klassenmerkmalen in einem Schritt erfolgt. Eine modellbasierte Auswertung von Punktwolken hingegen ist in verschiedenen Auswertestufen gut zu unterteilen. In jeder Stufe wird nach einem bestimmten und beschreibbaren Merkmal in der Punktwolke gesucht. Punkte, die das gesuchte Merkmal mit ähnlicher Ausprägung tragen, werden als Segmente oder Klassen zusammengefasst. Ein Beispiel hierfür ist die Unterteilung von Punktwolken in zwei oder mehrere Segmente in Abhängigkeit von der Distanz zum Aufnahmestandort (Merkmal ist hier die Distanz). Die erste Auswertungsstufe ist häufig das „Filtern“ von fehlerhaften Punkten, deren Auftreten in der Punktwolke zum Teil beschrieben werden kann. Dieser Stufe folgen verschiedene weitere Stufen, in denen Segmentierung- und

Klassifikationsverfahren mit dem Ziel der Generierung von Objektklassen angewendet werden.

Die graphbasierte Segmentierung von Punktwolken ist ein weit verbreitetes Verfahren, anhand dessen die Arbeitsschritte 1 bis 3 der Segmentierung gut nachzuvollziehen sind. Die einzelnen Punkte der Punktwolke stellen die Knoten des Graphen dar, die durch Kanten miteinander verbunden sind. Jede Kante erhält, aufgrund der Merkmalsunterschiede zwischen den Punkten, ein oder mehrere Gewichte, die gemessen oder berechnet werden (1). STORM ET AL., (2010) nutzen z. B. Farbinformationen (RGB-Werte), euklidische Distanzen und die Richtung der Punktnormalen, die über ein lokales Netz berechnet werden. Weitere Merkmale, die Laserscanner messen und für die Unterscheidung von Objekten einen Mehrwert darstellen, sind die Intensität oder die Rückkehrreihenfolge des empfangenen Signals. Bei der graphbasierten Segmentierung werden die Kanten mit den dazugehörigen Punkten anhand der Gewichte eines Merkmales, i. d. R. absteigend, sortiert (2) und ein Startgrenzwert für jedes Gewicht wird festgelegt (3). In einem iterativen Prozess werden Punkte einem Segment zugeordnet, Segmente zusammengefasst oder neue Segmente erstellt. Die Entscheidung, ob Segmente zusammengefasst oder neue Segmente gebildet werden, wird durch einen Grenzwert oder durch alle Grenzwerte bestimmt (FELZENSZWALB & HUTTENLOCHER, 2004). Erweiterungen des Algorithmus sehen ein dynamisches Anpassen der Grenzwerte vor, um optimale und detaillierte Segmente zu berechnen. Dieses Verfahren wird häufig um Voxel-Gitter erweitert (wie bei AIJAZI ET AL., 2013), da bei unstrukturierten Punktwolken durch ein festes oder ein dynamisches Gitter die Auswertung vereinfacht und beschleunigt werden kann. Die graphbasierte Segmentierung kann auf verschiedenen Oberflächen, wie *vermaschten* Punktwolken, Voxel-Gittern oder Oberflächenmodellen, erfolgen. Bei der Erstellung dieser Modelle erfolgt immer eine Generalisierung der Messwerte, so dass eine punktscharfe Segmentierung, wie sie für die Klassifikation von Messfehlern notwendig wäre, nicht mehr möglich ist.

2.2 Klassifizierung mit ConvNet

ConvNets werden für die detaillierte und automatische Klassifikation von Bildern eingesetzt, um die Inhalte der Bilder automatisch zu entschlüsseln und diese nach semantischen Aspekten zu clustern (GIRSHICK ET AL., 2014, GIRSHICK, 2015, REN ET AL., 2016). In digitalen Bildern sind die Merkmale, die für die Klassifikation von Objekten verwendet werden, in gleichmäßigen und gleich großen Rastern (Pixel) angeordnet. Diese Anordnung der Merkmale und die scharfe Abgrenzung der Merkmale bei gleichzeitiger lückenloser Verfügbarkeit ermöglichen ein sofortiges und effizientes Verarbeiten der Bilder mit Verfahren der Matrizenrechnungen. Mittels der Merkmale in den Eingabebildern und dessen Nachbarschaft, werden neue multidimensionale Merkmale in einer Convolutional-Schicht extrahiert. Neue Merkmale werden durch das Multiplizieren der Information mit festen Gewichten, die in einem ein- oder mehrdimensionalen Filter angeordnet sind, bestimmt. Die Größe des Filters und die Gewichte, werden für die Klassifizierungsaufgabe so ausgewählt, dass eindeutige Merkmale bestimmt werden können (Abb. 1).

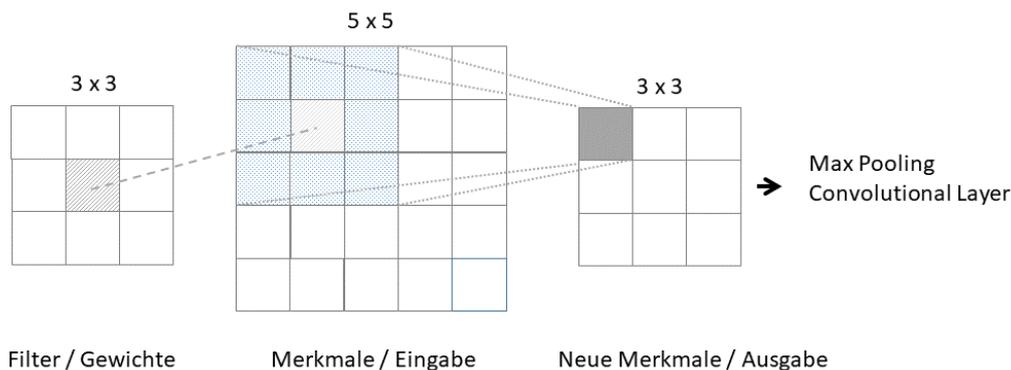


Abb. 1: Funktion einer Convolutional-Schicht am Beispiel einer 5 x 5 Eingabe und eines 3 x 3 Filters ohne Ausfüllen des Filters. Im Anschluss an diese Schicht können weitere Convolutional- oder Pooling-Schichten folgen.

Liegt für jeden Merkmalsträger mehr als ein Merkmal vor (dieses ist z. B. der Fall, wenn ein Bild aus drei Farbkanälen besteht), dann wird der Filter auf jeden Informationskanal angewendet und die Summe der neuen Merkmale je Träger bestimmt. Die Merkmale aus einer oder mehreren Convolution-Schichten werden durch das Pooling aggregiert. Hierbei wird ein weiteres Raster über eine feste Anzahl an Merkmalsträgern gelegt und die Merkmale werden zu einem Wert in einer Rasterzelle zusammengefasst. Hierfür wird meist der größte, der kleinste oder der mittlere Merkmalswert verwendet. Die eigentliche Klassifizierung wird durch ein „normales“, künstliches neuronales Netz (KNN) durchgeführt, in dem alle Merkmale der letzten Convolution-Pooling-Schicht mit den möglichen Netzausgaben (Klassen) verknüpft werden, so dass ein Vektor aufgestellt wird, der für jede Klasse eine Netzausgabe ausgibt. Funktionen wie Softmax, die auf dem Vektor angewendet werden, ermöglichen das Bestimmen der wahrscheinlichsten Klasse für jeden Merkmalsträger bzw. jedes Pixel (SZE ET AL., 2017).

2.3 ConvNet für Punktwolkenklassifikationen mit Gitterstrukturen

Punktwolken sind i. d. R. unsortiert, weisen regional unterschiedliche Punktdichten und eine unregelmäßige Verteilung der Merkmale auf, so dass viele Punktwolkenklassifizierungsverfahren einen Zwischenschritt benötigen. Dieser Zwischenschritt hat das Ziel, die Punktwolken in eine Struktur zu überführen, die Pixel oder Voxel nutzt. Hierfür werden die Punktwolken in andere Räume projiziert und Informationen zusammengefasst. Details gehen durch diese Vorverarbeitung verloren und fehlerhafte Punkte, die nur vereinzelt auftreten, werden bei der späteren Klassifizierung fälschlicherweise einer anderen Objektgruppe zugeordnet.

Anwendungen, bei denen größere einzelne Objekte in der Punktwolke während der Aufnahme zu klassifizieren bzw. durch eine *Bounding Box* zu markieren sind, sind aktuell nur durch eine Generalisierung der Punktwolke vor der Klassifizierung möglich. *PIXOR* (YANG ET AL., 2018) ist ein ConvNet für die Klassifizierung von Fahrzeugen und deren Bewegungsrichtung in dreidimensionalen Punktwolken. Hierfür wird eine Generalisierung durch das Erzeugen einer Vogelperspektivenansicht durchgeführt. Diese 2D-Ansicht wird in Voxel / Pixel unterteilt. Auf dieser vorverarbeiteten Punktwolke können die 2D-ConvNet angewendet werden.

Die Voxel-Struktur wird in einer Vielzahl von Arbeiten als Grundlage für ein *occupancy grid* verwendet. Ein *occupancy grid* ist eine Rasterstruktur, in der die Zellen dem Zustand „vorhanden sein“ oder „nicht vorhanden sein“ von Punkten zugeordnet werden. Durch dieses Raster werden die Punktwolken abstrahiert. Ziel der Verwendung des *occupancy grids* ist es, eine Punktwolke effizient in die 26 Klassen des *Sydney Urban Objects Dataset* (DEUGE ET AL., 2013) zu unterteilen. Bei *VoxNet* (MATURANA & SCHERER, 2015) wird die Punktwolke in quadratische Voxel-Segmente unterteilt. Jedes Voxel-Segment wird wiederum in 32^3 Subvoxel unterteilt, für die ein Wert für den Besetzungszustand (z. B. binärer oder als punktdichte Wert) berechnet wird. Jedes Voxel-Segment ist ein Tensor aus $32 \times 32 \times 32$ Einträgen. Dieser Tensor wird an ein 3D-ConvNet übergeben und für das Voxel-Segment wird eine Klasse bestimmt, der alle Punkte, die in dieses Segment fallen, zugeordnet werden.

HACKEL ET AL. (2017) folgen diesem Verfahren, berechnen aber um jeden Punkt der Punktwolke ein $16 \times 16 \times 16$ großes Voxel-Gitter bei fünf unterschiedlich großen Kantenlängen (von 2,5 bis 40 cm). Für jeden der Voxel wird ein Besetzungszustand berechnet, so dass ein $5 \times 16 \times 16 \times 16$ Tensor entsteht, der die geometrische Nachbarschaft des Punktes beschreibt. Die Merkmale dieses Tensors für jeden Punkt werden mit einem ConvNet in Anlehnung an das ConvNet *VGG* von SIMONYAN & ZISSERMAN (2014) verarbeitet, so dass für jeden Punkt die Klassifizierung durch den *Softmax*-Layer (Klassifizierungsfunktion) erfolgt. Dieser Ansatz, der punktorientierten Klassifizierung von komplexen dreidimensionalen Punktwolken, wird bei *PointNet* und dessen Erweiterungen weiterverfolgt.

2.4 PointNet und Erweiterungen

Das *PointNet* (QIET AL. 2017A) in seiner Grundform besteht aus einer Eingabeschicht, in der eine vollständige, kleine Punktwolke (2000 bis 4000 Punkte) oder ein Punktwolkensegment (Ausschnitt einer großen Punktwolke) als Tensor verarbeitet wird. Die Punktwolke bzw. der Tensor besteht mindestens aus den Punkten (n) mit Koordinatentripel (obligatorisch) und den optionalen Merkmalen (m), wie Normalen-Vektoren der Punkte, RGB- oder Intensitätswerten. Die Werte des eingelesenen Tensors werden durch ein *T-Net*, eine ConvNet für eine Starrkörpertransformation, in den Schwerpunkt des Punktwolkensegments transformiert. Diese Transformation kann sowohl auf Merkmale als auch auf Koordinaten angewendet werden. Nach der Transformation sind die Verarbeitungsschritte des *PointNet*, die hochdimensionale Merkmalsextraktion, die Sortierung und das Zusammenfassen von Merkmalen, so dass Punkte aufgrund der Merkmale einer Klasse zugeordnet werden können. Die Funktionsweise von *PointNet* entspricht dabei zweier verketteter Funktionen. Die innere Funktion extrahiert die Merkmale auf Grundlage der Merkmale der vorherigen Schichten. Dieses wird durch *Multilayer Perceptron* (MLP) erreicht. MLP sind mehr Schichten von verketteten Neuronen eines KNN. Die äußere Funktion ist die sortierende bzw. aggregierende Funktion, die Merkmale zusammenfasst. Diese wird durch eine *Max-Pooling-Schicht* umgesetzt. Für die Segmentierung bzw. punktweise Klassifikation werden lokale und globale Merkmale miteinander kombiniert. D. h. ein neuer Tensor mit den Dimensionen ($n \times m_{\text{lokal}} + m_{\text{global}}$), der aus den aggregierten Merkmalen und den lokalen Merkmalen jedes Punktes besteht, wird erstellt. Aus diesem Tensor werden wieder neue Merkmale je Punkt extrahiert und aggregiert. Für jeden Punkt werden Merkmale durch die MLP zusammengefasst, so dass eine Klassifizierung, in k vorgegebenen Klassen, erfolgen kann. Diese Klassifizierung erfolgt aufgrund des höchsten Wertes des Klassenvektors jedes Punktes (Abb. 2).

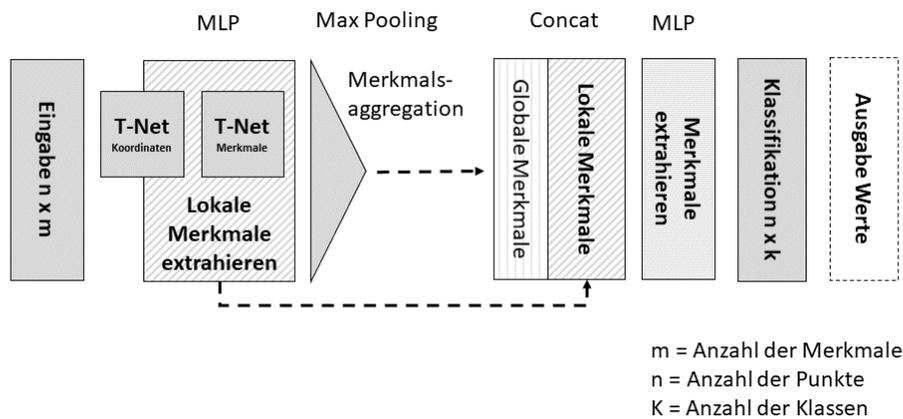


Abb. 2: Vereinfachte Darstellung des *PointNet* Verfahrens zur Segmentierung und Klassifizierung von Punktwolken in Anlehnung an QI ET AL., (2017A). *Multilayer Perceptron* (MLP) werden für die Extraktion von Merkmalen verwendet, die durch eine *Max Pooling* Funktion zusammengefasst werden. Die *Concat*-Funktion kombiniert Tensoren. Für jeden Punkt wird der höchste Ausgabewert aus den vorgegebenen Klassen bestimmt und so klassifiziert.

PointNet in dieser Grundform kann nur Merkmale nutzen, die im Punktwolkensegment vorhanden sind. Bei großen und unterschiedlich dichten Punktwolken führt dies zu fehlerhaften Klassifikationsergebnissen. QI ET AL., (2017B) nutzen *PointNet* als ein Baustein, führen aber eine Struktur von unterschiedlichen Schichten ein, in denen eine große Punktwolke schrittweise verkleinert wird (*PointNet++*). Aus der Punktwolke werden Punkte mit dem *farthest-point-sample (fps)* -Algorithmus ausgewählt, die das Zentrum einer Region bilden. Die Punkte, die zu dieser Region gruppiert werden, werden über einen festen Radius ausgewählt, wodurch die Größe der Region immer konstant ist, aber die Anzahl der Punkte variiert. Für jede Region wird auf Grundlage der Merkmale ein regionaler Merkmalsvektor durch *PointNet* berechnet, so dass für jede Region ein neuer Merkmalsvektor entsteht. Aus allen neuen Merkmalsvektoren werden in gleicher Weise in der folgenden Schicht neue Merkmalsvektoren berechnet. Wenn ein bestimmtes Abstraktionslevel für die Punktwolke, bzw. nun die Merkmale erreicht ist, dann werden diese Merkmale in der Segmentierungsphase wieder entschlüsselt. Schichtweise werden die Merkmale an die Zentralpunkte übertragen. Die Merkmale werden durch eine Interpolation an die benachbarten Punkte in der jeweiligen Schicht übertragen, so dass alle Punkte einen Merkmalsvektor mit ihren Merkmalen der zwei vorherigen Schichten haben. Mittels eines *PointNet* Bausteins werden für jeden Punkt aus diesen Merkmalen neue Merkmale aggregiert. In der letzten Segmentierungsschicht hat jeder Punkt ein Set an Merkmalen, welches für die Klassifizierung jedes Punktes verwendet wird. Hierbei liegt die Annahme vor, dass sich Merkmale gleichmäßig ausbreiten, was aber bei Punktwolken mit heterogenen Objektvorkommen nicht zwangsläufig ist. *PointNet++* kann um Berechnungsschritte, die die unterschiedliche Punktdichte berücksichtigen, erweitert werden.

ENGELMANN ET AL., (2017) adressieren ebenfalls das *PointNet* Problem, dass keine Merkmale außerhalb eines Punktsegmentes geteilt werden und optimieren die Klassifikationsleistung durch das Teilen von Merkmalen zwischen benachbarten Punktsegmenten. Dieses entspricht der natürlichen Merkmalsausbreitung in Punktwolken mit einer Vielzahl von unterschiedlichen Objekten. In dieser Erweiterung werden zwei Prozessketten vorgestellt, die zum einen die Eingabeschicht und zum anderen die Aggregation von Merkmalen verschiedener

Punktsegmente betreffen. In der ersten Prozesskette werden Punktsegmente mit verschiedenen Größen um einen Punkt gebildet, die unabhängig mit einem *PointNet* Baustein ausgewertet werden. In einer Verdichtungsschicht werden die Merkmalsvektoren der einzelnen Blöcke vereinigt und durch eine *Max-Pooling-Schicht* aggregiert. Die aggregierten Merkmale und lokalen Merkmale werden wieder verkettet (*concat*) und für alle Punkte wird ein $n \times k$ Tensor aufgestellt, mit dem die Klassifizierung für jeden Punkt berechnet wird.

Die zweite Prozesskette sieht vor, dass eine feste Anzahl an benachbarten Punktsegmenten gleichzeitig Merkmale extrahiert und diese teilt. Die Aggregation der Merkmale erfolgt durch ein *Recurrent* (wiederholendes) Neuronales Netz (RNN). Mit dieser Art des KNN werden Informationsketten verarbeitet, um neue Merkmale über einen Abschnitt oder die gesamte Informationskette zu erhalten (GOODFELLOW ET AL., 2016, S 368ff). Diese aggregierten Merkmale werden Punktsegmentweise zusammengefasst und für alle Punkte aller Punktsegmente erfolgt eine Klassifikation separat.

3 Einfluss von Trainingsdaten

Die Klassifizierungsleistung von ConvNet wird neben der Art und Weise, wie die Merkmale extrahiert und aggregiert werden (Netzwerkarchitektur), maßgeblich von den Daten, mit denen das Wissen erlernt wird (Trainingsdaten), beeinflusst. Für eine zuverlässige Klassifizierung einer Punktwolke ist das ConvNet mit Trainingsdaten zu trainieren, in denen ausreichend viele Punkte mit ähnlichen Merkmalen vorhanden sind, damit die Klassen der Punktwolke zuverlässig erlernt werden können. Als Einschränkung zu dieser Forderung müssen die Merkmale aber eine Verschiedenheit bzw. Streuung innerhalb der Klassen aufweisen, damit auch eine Klassifizierung erfolgen kann, wenn Objekte in einem anderen szenischen Zusammenhang auftreten, ein Messrauschen vorliegt oder fehlerhaft gemessene Punkte in der Punktwolke vorhanden sind. Der Einfluss von fehlerhaft gemessenen Punkten ist ein noch wenig untersuchtes Themenfeld.

In BARNEFSKE & STERNBERG (2019) konnte beobachtet werden, dass Punktwolken, in denen fehlerhafte Punkte vorhanden sind, weniger zuverlässig klassifiziert werden, als Punktwolken, in denen diese Punkte entfernt wurden. Der Einfluss dieser fehlerhaften Punkte, die vornehmlich durch Brechung und Reflexion an Glas- und Spiegelflächen, der Strahlendivergenz an Ecken und Kanten oder beweglichen Objekten in der Aufnahmeszene stammen, wird nun anhand von zwei Punktwolkensets näher untersucht. Für die Untersuchungen wird die Punktwolken von mehreren Standpunkten des Punktwolkensets mit *PointNet* trainiert und evaluiert.

3.1 Netzwerkarchitektur, Trainingsparameter und Optimierungsmethode

Die Untersuchung der Klassifizierungsleistung von Punktwolken, die aus verschiedenen Klassenkombinationen bestehen, wird mit der Netzwerkarchitektur *PointNet* für die semantische Klassifikation von Szenen durchgeführt. Hierfür wird *PointNet* mehrfach mit den gleichen Trainingsparametern trainiert und die Evaluationsergebnisse werden nach 50 Epochen verglichen. Für das Training wird, in Anlehnung an QI ET AL., (2017A), die *Adam*-Optimierungsfunktion und eine *Batchsize* von 20 verwendet. Alle Gewichte des Netzes werden für jeden Trainingsdurchgang zufällig initialisiert. Die Blöcke der Eingabeschicht beinhalten

4096 Punkte und bestehen nur aus geometrischen Merkmalen, in Form von den Koordinatentripeln. Diese Restriktion auf 4096 Punkte je Block ist notwendig, damit die Berechnungen auf einem *Nivida GeForce GTX 1080 GPU* durchgeführt werden können.

3.2 Trainings- und Untersuchungspunktwolken

Die Klassifizierungsleistung wird mit Punktwolken untersucht, die Straßen und einen Park in der Hamburger HafenCity zeigen. Die Punktwolken wurden mit zwei verschiedenen Laserscannern an denselben Aufnahmeorten erstellt, so dass die gleichen Objekte in jeder Punktwolke vertreten sind. Variationen treten durch Fahrzeuge und Fußgänger auf. Die Punktwolken des ersten Sets wurden mit dem Laserscanner *Zoller + Fröhlich Imager5010* mit einer Auflösungseinstellung von 6 mm bei 10 m erfasst. In der Nachbearbeitung wurden die Punktwolken durch Panoramabilder eingefärbt, die an der gleichen Position erstellt wurden. Das zweite Punktwolkenset wurde mit dem *Faro Focus3D* Laserscanner erstellt. Die Punktdichte in einer Entfernung von 10 m ist ähnlich zu der des ersten Punktwolkensets. Die Kolorierung der Punktwolke erfolgt durch die integrierte Kamera des Laserscanners.

Alle Punktwolkensets wurden manuell in elf Klassen (Straße, Gehweg, Bodenvegetation, Mensch, Stativ, Auto, Bauwerk, Baum, Straßenschild, fehlerhafte Punkte und Sonstiges) eingeteilt und können für die folgenden Untersuchungsfragen kombiniert werden. Diese elf Klassen teilen die Punktwolke in die größten Sinnklassen und in die Klassen, die für die Untersuchung von besonderem Interesse sind, auf. Die Anzahl der Punkte in jeder Klasse variiert von weniger als 1 % (z. B. Stativ) bis zu 30 % (z. B. Bauwerk). Dies entspricht den normalen Verhältnissen in gemessenen Punktwolken, stellt aber für das Training eine ungünstige Kombination dar, da einige Klassen nur durch sehr wenige Punkte erlernt werden können. Es werden vier Kombinationen an Klassen erstellt:

- (A) Alle Klassen.
- (B) Alle Klassen, die mit mehr als 2 % an Punkten vertreten sind.
- (C) Alle Klassen aus (B) und die Klasse fehlerhafte (gemessene) Punkte.
- (D) Fehlerhafte (gemessene) Punkte und alle restlichen Klassen zusammengefasst als eine Klasse.

Die Anzahl der Punkte, die in einzelnen Punktwolken vorkommen, variiert in einigen Fällen sehr stark, so dass für das Training verschiedene Punktwolken kombiniert werden, damit die Anzahl der Punkte möglichst ausgeglichen ist. Die Evaluation erfolgt immer anhand derselben Punktwolke, so dass die Untersuchungen in sich vergleichbar sind, aber nicht zwingend auf andere Punktwolken übertragen werden können.

Die Klassifizierungsleistung von Punktwolken wird mittels einer Punktwolke evaluiert, die keine Überschneidung zu den Trainingspunktwolken aufweist, aber aus dem gleichen Umfeld stammt. Das bedeutet, dass die Bauwerke, die Bodenoberflächen und die Straßenschilder gleich sind. Für die Bewertung der Klassifizierungsleistung wird der Parameter *Intersection over Union* (IoU) verwendet. Die IoU ist die Schnittmenge der wahren und prädizierten Punkteklassen. Dieser Parameter setzt sich aus den zwei Parametern Recall und Präzision zusammen. Der Recall beschreibt das Verhältnis zwischen richtig klassifizierten Punkten und den wahren Punkten einer Klasse. Die Präzision drückt das Verhältnis zwischen richtig klassifizierten Punkten und Punkten, die fehlerhafterweise dieser Klasse zugeordnet wurden, aus.

3.3 Untersuchung der Klassifizierungsleistung

Die erste Klassenkombination (A) umfasst alle Klassen und zeigt die Klassifizierungsleistung für die gesamte Punktwolke. Alle Punkte der *Imager5010* Punktwolke werden in eine von vier Klassen, und zwar Baum, Bauwerk, Straßenschild oder fehlerhafte Punkte, eingeordnet. Die Klassifizierung von Bodenoberflächen ist für diese Klassenkombination nicht möglich, auch werden die kleinen Klassen nicht berücksichtigt. Die Werte des Recall und der Präzision von kleiner 50 % sowie einer maximalen IoU über alle Klassen von 10 % deuten darauf hin, dass für diese Klassenkombination keine zuverlässige Klassifizierungsleistung erzielt werden kann (Tabelle 1).

Alle Punkte der *Focus3D* Punktwolke werden bei allen Klassifizierungsdurchgängen nur der größten Klasse Bauwerk zugeordnet (Tabelle 1), so dass festzustellen ist, dass eine Klassifizierung mit *PointNet* für diese Punktwolke nicht gelingt.

Tabelle 1: Klassifikationsergebnis bei Verwendung aller Klassen. b = beste Klassifizierung und s = schlechteste Klassifizierung. Den Klassen Straße, Gehweg, Mensch, Sonstiges, Stativ, Auto und fehlerhafte Punkte wurden keine Punkte zugewiesen.

Punktwolken	Recall in %				Präzision in %				IoU in %			
	Baum	Bauwerk	Straßens.	f. Punkte	Baum	Bauwerk	Straßens.	f. Punkte	Baum	Bauwerk	Straßens.	f. Punkte
Imager5010												
(b)	26	35	16	29	47	49	2	4	20	26	2	4
(s)	12	44	0	34	37	47	0	4	10	29	0	4
Focus3D												
(b)	0	100	0	0	0	33	0	0	0	33	0	0

QIET AL., 2017A klassifizieren mit *PointNet* eine photogrammetrische Innenraumpunktwolke des Datensets *3D Semantic Parsing of Large-Scale Indoor Spaces* in 13 Klassen und erreichen eine IoU von 48 %. In dieser Punktwolke existiert keine Klasse von fehlerhaften Punkten. Die Verhältnisse der Klassengrößen werden nicht näher beschrieben bzw. sind nicht Gegenstand von Untersuchungen. Zudem sind diese Punktwolken durch ein anderes Messsystem aufgenommen worden. Im Folgenden wird u. a. anhand der Kombinationen B und C untersucht, welchen Einfluss die Klassengrößen (B) und die Klasse: fehlerhafte Punkte (C) auf die Klassifizierungsleistung von LIDAR-Punktwolken, die mit *PointNet* klassifiziert werden, haben.

Auffällig bei der Klassenkombination B ist, dass es zu einer Verwechslung bei den Klassen der Bodenoberflächen kommt. In der *Imager5010* Punktwolke wird die Straße nicht richtig klassifiziert und die Klassen: Gehweg (Recall 95 %) und Bodenvegetation (Recall 59 %) gut klassifiziert. Eine gegensätzliche Beobachtung kann für die *Focus3D* Punktwolke gemacht werden. Hier wird die Klasse: Straße mit bis zu 99 % Recall klassifiziert und die Klassen: Gehweg und Bodenvegetation schwach und nicht klassifiziert (Tabelle 2). Es ist festzustellen, dass es bei diesen drei Klassen mit einer ähnlichen geometrischen Ausprägung zu einer

Überklassifizierung einer Klasse bzw. Minderklassifikation der anderen Klassen kommt. Die Klassen: Bauwerk und Baum, die ein Volumen ausfüllen können, können für beide Punktwolken mit höheren *Recall* klassifiziert werden. Ein möglicher Grund für diese Beobachtung ist, dass das *PointNet* für Klassifizierungen von 3D-Objekten entwickelt wurde und dass für die Klassifizierung in diesen Untersuchungen nur geometrische Merkmale verwendet wurden. Eine Unterscheidung von ebenen Klassen ist mit diesen Daten besonders ungünstig.

Tabelle 2: Klassifikationsergebnis unter Verwendung der sechs Objektklassen, die mehr als 2 % der Punkte repräsentieren. b = beste Klassifizierung und s = schlechteste Klassifizierung.

Punktwolken	Recall in %						Präzision in %						IoU in %					
	Straße	Baum	Bauwerk	Straßens.	Bodenv.	Gehweg	Straße	Baum	Bauwerk	Straßens.	Bodenv.	Gehweg	Straße	Baum	Bauwerk	Straßens.	Bodenv.	Gehweg
Imager5010																		
b	0	92	89	1	59	95	0	90	87	3	40	53	0	87	79	1	31	52
s	0	94	67	6	0	94	0	71	75	24	0	53	0	68	55	5	0	51
Focus3D																		
b	99	70	72	40	0	0	46	69	87	6	0	0	46	53	65	5	0	0
s	82	88	72	84	0	17	43	73	95	18	0	51	39	67	70	17	0	15

Die Untersuchungen mit der Klassenkombination B zeigen des Weiteren, dass Klassifizierungsleistung mit einer IoU von 57 % für den *Imager5010* und von 51 % für den *Focus3D* erzielt werden können. Diese Klassifizierungsleistung ist mit den Ergebnissen QI ET AL., (2017A) vergleichbar, so dass festgestellt werden kann, dass das Messsystem, mit dem eine Punktwolke aufgenommen ist, nicht zwangsmäßig einen Einfluss auf die Klassifizierungsleistung von *PointNet* hat. Eine Steigerung der Klassifizierungsleistung wird hervorgerufen, wenn a) keine sehr kleinen Klassen oder / und b) keine fehlerhaften Punkte in der Punktwolke vorhanden sind. Zur Klärung, welche dieser Einflussgrößen maßgeblich verantwortlich sein kann, werden Klassifizierungen mit der Kombination C durchgeführt.

Die Klassenkombination C umfasst zusätzlich zu den Klassen aus B, die Klasse der fehlerhaften Punkte. Mittels dieser Untersuchung wird überprüft, ob das Vorhandensein dieser Klasse die Klassifizierungsleistung beeinflusst, welches für die *Imager5010* Punktwolke festgestellt werden kann. Für diese Punktwolke erfolgt eine Klassifizierung, wie bei der Klassenkombination A, nur in die gleichen vier Klassen. Die Werte für Recall und Präzision sind mehrheitlich um nur wenige Prozent höher, so dass eine signifikante Steigerung, aufgrund des Einflusses der Klassengrößen, hier nicht bestätigt werden kann (Tabelle 3).

Eine mögliche Strategie, den Einfluss von fehlerhaften Punkten zu eliminieren, kann sein, die Klassifikation in mehreren Stufen, ähnlich den modellbasierten Klassifizierungsverfahren, durchzuführen. In der ersten Stufe werden nur fehlerhafte Punkte von allen übrigen Objektklassen durch ein ConvNet getrennt. In den anschließenden Stufen wird die Punktwolke in Objektklassen unterteilt. Zur Untersuchung, ob mit *PointNet* eine Vorklassifikation durch-

geführt werden kann, werden Punktwolken in die zwei Klassen: fehlerhafte Punkte und Objekte aufgeteilt (Kombination D) und es wird mit *PointNet* trainiert. Es ist anzumerken, dass der Anteil der fehlerhaften Punktwolken unter den Laserscannern unterschiedlich groß ist. Für die Punktwolke des *Imger5010* sind 6,9 % der Punkte fehlerhafte Punkte und für den *Focus3D* 4,0 %. Für die Punktwolken des *Focus3D* konnte eine Unterscheidung zwischen Objekten und fehlerhaften Punkten nicht erreicht werden. Alle Punkte werden der anzahlmäßig stärksten Klasse zugeordnet, welches immer zu einer IoU von 96,0 % bzw. 99,9 % führt. Eine Differenzierbarkeit ist nur bei den Punktwolken des *Imager5010* zu beobachten, die zwischen 28,0 % und 80,0 % *IoU* stark variiert (Tabelle 4). In dieser Untersuchung zeigt sich erneut, dass das Verhältnis zwischen der Punktzahl bei verschiedenen Klassen, die Klassifizierung beeinflusst.

Tabelle 3: Klassifikationsergebnis unter Verwendung der sechs Objektklassen, die mehr als 2 % der Punkte repräsentieren und der Klasse: fehlerhafte Punkte. Mit b = beste Klassifizierung und s = schlechteste Klassifizierung. Die Klassen für Bodenflächen: Straße, Bodenvegetation und Gehweg wurden durch *PointNet* nicht klassifiziert.

Punktwolken	Recall in %				Präzision in %				IoU in %			
	Baum	Bauwerk	Straßens.	f. Punkte	Baum	Bauwerk	Straßens.	f. Punkte	Baum	Bauwerk	Straßens.	f. Punkte
Imager5010												
(b)	10	54	4	5	37	24	1	2	8	20	1	2
(s)	8	29	15	28	41	89	1	4	7	28	1	3

Tabelle 4: Klassifikationsergebnis bei einer Aufteilung der Punktwolke in die zwei Klassen: fehlerhafte Punkte und Objekte. Mit (b) beste Klassifizierung und (s) schlechteste Klassifizierung. Für den *Focus3D* wurden immer dieselben Ergebnisse erzielt.

Punktwolken	IoU in %			Verteilung in Punkte in %	
	alle	Objekte	fehlerh. Punkte	Objekte	fehlerh. Punkte
Imager5010 (b)	80	89	8	93,1	6,9
Imager5010 (s)	28	43	3	93,1	6,9
Focus3D	92	96	0	96,0	4,0

3.4 Umgang mit den Einflussgrößen

Die Klassifizierungsuntersuchungen zeigen, dass der Inhalt und die Form der Trainingspunktwolken (z. B. die Klassenaufteilung, in der die Punktwolken für das Training bereitgestellt werden) einen Einfluss auf die Klassifizierung haben. Die Netzwerkarchitektur *PointNet* ohne Erweiterungen wurde in dieser Arbeit angewendet und nicht verändert.

ENGELMANN ET AL. (2017) zeigen, dass für synthetische Punktwolken durch einfache Erweiterungen an der Eingabeschicht und der Aggregation Klassifizierungen mit einem IoU von 90 % möglich sind. Ähnliche Adaptionen und die Vorschaltung von weiteren Vorverarbeitungsnetzwerken sind zentrale Parameter für die Entwicklung eines praxistauglichen Klassifizierungsverfahrens.

Die in dieser Arbeit untersuchten Einflussgrößen betreffen die Trainingsdaten. Hierbei wurde das Vorhandensein von fehlerhaften Punkten, die Anzahl der Klassen und das Punkteverhältnis bei verschiedenen Klassen untersucht. Fehlerhafte Punktwolken sind aufgrund der Untersuchungsergebnisse einflussgebend für Klassifizierungsleistung. Eine einfache Klassifizierung nach fehlerhaften und nicht fehlerhaften Punkten ist, aufgrund der Ergebnisse für die Klassenkombination D, nicht möglich. Es ist hier zu untersuchen, ob bei einem günstigeren Verhältnis von fehlerhaften und nicht fehlerhaften Punkten beim Training eine Unterscheidung möglich ist.

Eine Optimierung der Klassifizierung bei einer großen Anzahl von Klassen mit stark variierender Punktanzahl ist durch eine Verkettung von ConvNet, die verschiedene Teilklassifizierungen durchführen, zu untersuchen. Unabhängige ConvNet führen zunächst eine Grobklassifizierung durch, die immer feinmaschiger wird. Beispielsweise werden Bauwerke von den Bodenflächen erst getrennt und in einem folgenden Schritt wird mittels ConvNet nach Klassen für Bodenfläche klassifiziert. Merkmale, die für eine Unterscheidung notwendig sind, könnten ggf. schneller und zuverlässiger detektiert werden.

4 Fazit und Ausblick

Datenbasierte Verfahren, wie ConvNet, bieten neue Möglichkeiten für eine detaillierte und effiziente *end-to-end* Punktwolkenklassifikation. Die Leistung dieser Verfahren ist u. a. von der Form und dem Inhalt der Trainings- und Anwendungsdaten abhängig. Je strukturierter die Daten vorliegen, desto effizienter sind die Trainings- und die Anwendungsphasen. Können Daten nicht in eine Datenstruktur überführt werden (z. B. Verlust von relevanten Details), dann ist die Auswertung aufwändiger und fehleranfälliger. Am Beispiel von *PointNet* wurde der Einfluss von fehlerhaften Punkten auf die Klassenaufteilung untersucht. Es konnte gezeigt werden, dass fehlerhafte Punkte die Klassifikation bei unstrukturierten Punktwolken negativ beeinflussen. Unterschiedlich große Anzahlen an Punkten je Klasse beeinflussen zudem die Klassifizierung, da bei sehr großen Unterschieden die Klasse, die die meisten Daten repräsentiert, überklassifiziert wird. Ungünstigen Konstellationen in den Trainingsdaten kann mit einem Anpassen der Trainingsdatengröße und dem Inhalt der Klassen sowie ggf. mit einer Verkettung von ConvNet begegnet werden.

Literatur

AIJAZI, A., CHECCHIN, P. & TRASSOUDAIN, L. (2013): Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach with Evaluation. *Remote Sensing*, 5, 4, 1624–1650.

- BARNEFSKE, E. & STERNBERG, H. (2019): PCCT: A Point Cloud Classification Tool To Create 3D Training Data To Adjust And Develop 3D ConvNet. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4216, 35-40.
- DEUGE, M. D., QUADROS, A., HUNG, C., DOUILLARD, B. (2013): Unsupervised Feature Learning for Classification of Outdoor 3D Scans. *Australasian Conference on Robotics and Automation*.
- ENGELMANN, F., KONTOGIANNIA, T., HERMANS, A. & LEIBE, B. (2017): Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 716-124.
- FELZENSZWALB, P. F. & HUTTENLOCHER, D. P. (2004): Efficient Graph-based Image Segmentation. *International Journal of Computer Vision*, 59, 2, 167–181.
- GIRSHICK, R. (2015): Fast R-CNN. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1440–1448.
- GIRSHICK, R., DONAHUE J., DARRELL, T. & MALIK, J. (2014): Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.
- GOODFELLOW, I., BENGIO, Y. & COURVILLE, A. (2016): *Deep Learning*. The MIT Press.
- HACKEL, T., SAVINOV, N., LADICKY, L., WEGNER, J.D., SCHINDLER, K. & POLLEFEYS, M. (2017): SEMANTIC3D.NET: A New Large-scale Point Cloud Classification Benchmark. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 91–98.
- MATURANA, D. & SCHERER, S. (2015): VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 922–928.
- QI, C. R., SU, H., MO, K. & GUIBAS, L. J. (2017A): PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 77-85.
- QI, C. R.; YI, L.; SU, H. & GUIBAS, L. J. (2017B): PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 5099-5108
- REN, S., HE, K., GIRSHICK, R. & SUN, J (2016): Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, 91–99.
- SIMONYAN, K. & ZISSERMAN, A. (2014): Very Deep Convolutional Networks for Large-scale Image Recognition, *Proceedings of the ICLR*, 1409-1556.
- STROM, J., RICHARDSON, A. & OLSON, E. (2010): Graph-based Segmentation for Colored 3D Laser Point Clouds. *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2131-2136.
- SZE, V., CHEN, Y.-H., YANG, T.-J. & EMER, J.S. (2017): Efficient Processing of Deep Neural Networks: A tutorial and survey. *Proceedings of the IEEE*, 105, 12, 2295-2329.
- YANG, B., LUO, W. & URTASUN, R. (2018): PIXOR: Real-Time 3D Object Detection from Point Clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7652–7660.

A.2 Evaluating the Quality of Semantic Segmented 3D Point Clouds

Reference:

Barnefske, E.& Sternberg, H. (2022): Evaluating the Quality of Semantic Segmented 3D Point Clouds. Remote Sensing, 14, 446. DOI: 10.3390/rs14030446

Graphical Abstract:

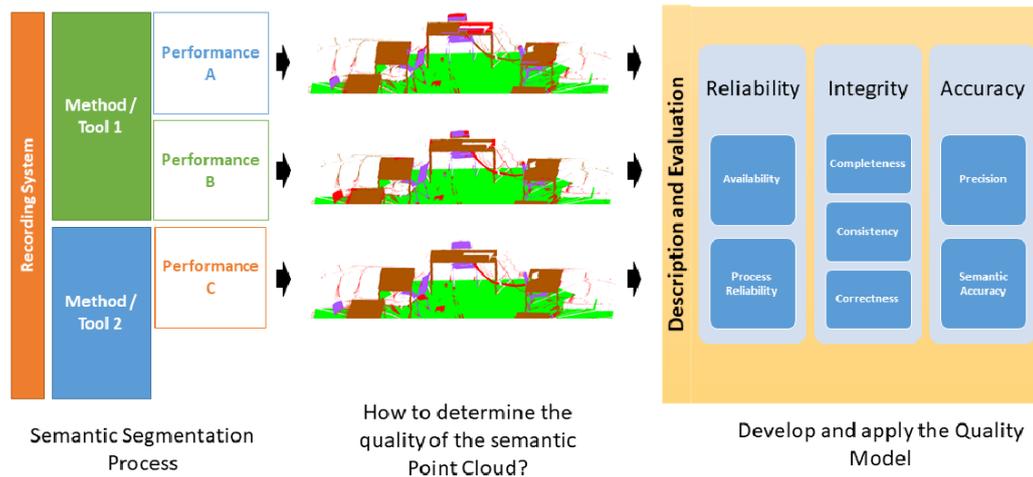


Figure 42: Graphical Abstract: Evaluation of semantic segmentation methods using the quality model.

Contribution of Co-Authors:

Table 7: Contribution to Paper No. 2

Involved in	Estimated contribution
Ideas and conceptual design	90%
Computation and results	100%
Analysis and interpretation	95%
Manuscript, figures and tables	100%
Total:	96%

I hereby confirm the correctness of the declaration of the contribution of Eike Barnefske for Paper 2 in Table 7:

Prof. Dr.-Ing. Harald Sternberg, HafenCity Universität Hamburg



Article

Evaluating the Quality of Semantic Segmented 3D Point Clouds

Eike Barnefske * and Harald Sternberg

Department of Hydrography and Geodesy, HafenCity University Hamburg, Henning-Voscherau-Platz 1, 20457 Hamburg, Germany; harald.sternberg@hcu-hamburg.de

* Correspondence: eike.barnefske@hcu-hamburg.de

Abstract: Recently, 3D point clouds have become a quasi-standard for digitization. Point cloud processing remains a challenge due to the complex and unstructured nature of point clouds. Currently, most automatic point cloud segmentation methods are data-based and gain knowledge from manually segmented ground truth (GT) point clouds. The creation of GT point clouds by capturing data with an optical sensor and then performing a manual or semi-automatic segmentation is a less studied research field. Usually, GT point clouds are semantically segmented only once and considered to be free of semantic errors. In this work, it is shown that this assumption has no overall validity if the reality is to be represented by a semantic point cloud. Our quality model has been developed to describe and evaluate semantic GT point clouds and their manual creation processes. It is applied on our dataset and publicly available point cloud datasets. Furthermore, we believe that this quality model contributes to the objective evaluation and comparability of data-based segmentation algorithms.

Keywords: 3D point cloud; quality model; annotation tools; datasets; evaluation metric; evaluation parameter



Citation: Barnefske, E.; Sternberg, H. Evaluating the Quality of Semantic Segmented 3D Point Clouds. *Remote Sens.* **2022**, *14*, 446. <https://doi.org/10.3390/rs14030446>

Academic Editor: Sander Oude Elberink

Received: 20 December 2021

Accepted: 13 January 2022

Published: 18 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A major research topic in geodesy is to digitize activities in construction [1–3], in building maintenance [4,5] and in navigation [6,7]. For the digitization of these tasks, digital building parts and furnishing objects must be formed and processed. Digital models of real-world buildings (digital twins) are needed to make complex and large semantic data interpretable for humans and machines [8]. The creation of digital twins is often based on 3D point clouds, which are efficiently captured with depth imaging cameras or light imaging, detection and ranging (LIDAR) systems. The 3D point cloud without any semantic features can already be considered a model, since humans can use their knowledge to interpret semantic point groups as single objects. These semantic point groups are, e.g., the objects and scanning artifacts, as shown in Figure 1.

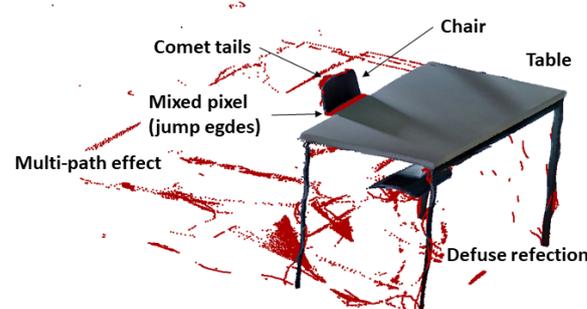


Figure 1. Examples of objects (chair and table) and scanning artifacts in a point cloud. Common scanning artifacts are: comet tails, mixed pixels on edges (jump edges), multi-path effects and defused reflections.

For the digital processing of point clouds, semantic information has to be given to the point cloud to form semantic segments. The initial semantic segmentation is always performed by humans. For this purpose, different tools can be used to form segments as efficiently, reliably, precisely and correctly as possible and to assign the correct semantic label. The efficiency, reliability, precision and correctness of semantic segmentation are characteristics that describe the quality of a semantic point cloud. These characteristics build the quality model, which describes how well the creation of the semantic point cloud works. Evaluation metrics now become parameters of the quality model, which describe the point cloud characteristics. A comparison of different segmentations is possible with the quality parameter. Method comparisons are common in automatic semantic segmentation [9–12], which typically uses machine learning (ML) and artificial intelligence (AI). For method comparisons, point cloud benchmarks are used [13,14]. Semantic point cloud benchmarks are point clouds for which a semantic ground truth (GT) is given. It is assumed that the GT point clouds are free of semantic and geometric errors. However, unfortunately, in most cases, a complete evaluation of the manually or semi-automatically created semantic point cloud benchmarks is not performed. The characteristics of a semantic point cloud that can be evaluated vary strongly among the published point clouds. In some works, the semantic accuracy of a point cloud is evaluated completely [13] or by spot checks [14,15]. Other works evaluate only the completeness and correctness of a building model [16]. Even if some characteristics of the point cloud can be evaluated, then a comparison of the evaluation metric is often not possible, since no uniform metrics are defined. For example, intersection over union (IoU), F1-score, overall accuracy, recall, precision and many others are used to validate the accuracy. The variety problem of the evaluation metric for the case of object detection in images is well known and a tool to translate the evaluation metrics for compression was developed [17].

To the best of our knowledge, a holistic quality model in which availability, integrity and accuracy are represented does not exist for semantic point clouds. Such a quality model has the potential to make the investigation of existing and upcoming GT point cloud datasets comparable. Deviation from reality, the availability of information and applicability to a certain purpose can be determined with our quality model for indoor point clouds.

Fundamental for the development of the quality model is the definition of the semantic segmentation, as well as its separation into detection and classification (Section 2.1). The capture methods of 3D point clouds for indoor applications (Section 2.2), the existing point cloud datasets (Section 2.3), as well as the tools for manual and semi-automatic semantic segmentations (Section 2.4) determine the characteristics needed in the quality model. The development of the quality model is derived from a process description (Section 3.1), a class definition (Section 3.2) and a data model (Section 3.3). The quality characteristics and parameters are defined and discussed in Section 3.4. The descriptive and evaluative use of the quality model is presented and discussed based on different point clouds in Sections 4.1 and 4.2. Finally, Section 5 summarizes the main conclusions and gives an outlook for further development and possible use of the quality model.

2. State of the Art

The surfaces of real objects are often represented as 3D point clouds after digitization. These 3D point clouds are an unsorted list of coordinates with additional (spectral) information. This representation is particularly well suited for measuring systems that use high-frequency scanning of object surfaces. Very efficient storage of single points or point groups (lines or arrays) is thus possible. This has caused the point cloud to become a quasi-standard for 3D object representations. The point cloud represents very efficiently, accurately and with a high resolution the geometry of scenes and objects. Unfortunately, with point clouds, the separation of individual objects is not possible right away. Thus, it is a necessary next processing step to derive information or models from point clouds.

Current research on the separation of point clouds is mainly applied to autonomous operating systems, building modeling and computer vision (CV) tasks. Autonomous operating systems include autonomously driving cars, where information for obstacle avoidance, route planning and sign recognition has to be generated from the 3D point clouds [18,19]. CV and building modeling aim to enrich the point cloud with semantic information. The enriched point clouds are the basis for decision making and the creation of semantic models. If the point clouds represent complex scenes in which individual objects appear several times, then instancing is often the goal. Applications include the modeling of digital twins or the creation of city models, as well as the direct creation of simple building models based on point clouds and prior knowledge [20–22].

Different types of acquisition systems, segmentation tools and semantic point cloud datasets are available, forming the basis for the development of automatic point cloud separation methods. The application of these sets the quality of a semantic point cloud. A large amount of semantic training and benchmark point clouds are available.

2.1. Classification, Object Detection and Segmentation

The definition of classification, object detection and segmentation is not clear in the literature, and these terms vary by research and application field. Different terms are used for the same separation task, or the meaning of the terms may be ambiguous. Some reviews [23,24] distinguish between classification, object detection and segmentation. Other researchers [25] use segmentation as an all-encompassing term for various categorization methods. To avoid misunderstandings, classification, object detection as well as semantic and instance segmentation are briefly defined below for this work.

Classification: Classification is the assignment of a class feature (label) to one object. This can be a single point, a point cloud, a segment of a point cloud or another geometry type. Usually, semantic labels or IDs are assigned. The classification in the following is understood as the assignment of one semantic label to one point cloud segment.

Object detection: In object detection, specific objects are defined based on geometric or spectral features in the point clouds. The individual object and not the entire point cloud is of interest, so that large parts of the point cloud are not evaluated in detail. Several objects in a point cloud can be detected and a unique identifier is obtained. Object detection is often used in conjunction with tracking objects in applications with multiple sub-point clouds. The objects are usually roughly described in terms of geometric size, position and orientation using bounding boxes. In other cases, it is not the objects as a whole that are of interest, but only certain surfaces or shapes [26]. These are searched for in the point clouds (shape detection).

Semantic segmentation: The semantic segmentation has the goal of extending the features of the points by semantic labels. Semantic labels are semantic classes that usually describe real-world objects. The difference for the classification is that the segments are formed in this process step and a label is set for all points of the segment. A semantic segment can consist of several geometrically independent segments. For example, a point can belong to the class *table*; complementarily, it can belong to the subclass *table leg*. Moreover, the results of the classification of each point can form a new segment.

Instance segmentation: An instance segment describes the geometric shape of one object. Instances in a point cloud can be distinguished by a unique identifier. An instance is usually enriched with semantic information. Points of the same semantic segment describe different objects. For example, if two tables are in one point cloud, then both carry the same semantic label. In order to distinguish the tables, instances must be created. Each table is an instance, which usually consists of a geometrically connected point cloud segment.

The creation of a digital twin goes beyond this idea. For modeling a digital twin, new parametrized objects have to be formed that describe the point cloud content by generalizations such as a simple geometry.

2.2. Captured and Synthetic Point Clouds

Almost any semantic 3D point cloud is derived from a synthetic surface model or is captured by contactless sensors. An overview of the methods is given in Figure 2.

Synthetic 3D point clouds are mostly generated from large collections of online model databases, such as [27]. These point clouds are generated efficiently by transforming a surface model into a regular or random point cloud. These points lie on the surface of the previous model or have synthetic noise added. Synthetic 3D point clouds usually represent only a single object or a small group of objects. Usually, they are used for algorithm development or prototype testing [28,29].

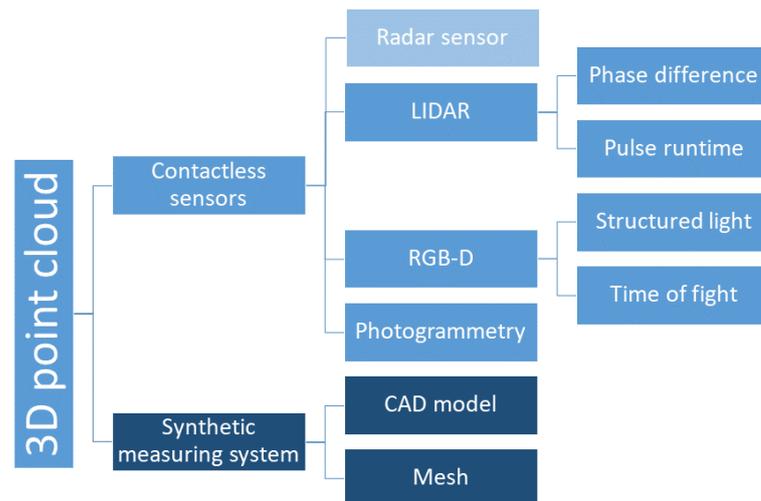


Figure 2. Capturing systems and basic data for the creation of 3D point clouds.

Any acquisition technique for capturing reality has a certain resolution, precision and correctness, which can be found in the resulting point cloud. These point cloud characteristics often depend on the surface of the object, the acquisition distance, the environmental conditions and the measurement sensors.

Optical sensors are the most widely used method for mapping reality. Optical sensors use light of different spectral bands to create a 3D point cloud of real environments with photogrammetric methods, as described in [30]. In particular, depth imaging cameras and LIDAR systems have been widely used in the last decade to create point cloud datasets [23–25]. The reasons are user friendliness, mainly moderate acquisition and evaluation costs [31,32] and the efficient capturing of larger areas. In addition to optical sensors, radar is sometimes used to create point clouds [33,34].

Depth imaging cameras consist of one or more cameras for different spectral ranges and an active emitter. Different principles for determining the image depths are used. For example, the *Matterport Pro 3D Camera* and the *Microsoft Kinect V1* use structured light (SL) [35] and the *Microsoft Kinect V2* uses the time of flight (ToF) method [36].

With the SL cameras, a monochrome near-infrared (NIR) image is captured in addition to a true-color image (red, green, blue (RGB)). The scene to be captured is illuminated by a projector with a known NIR pattern. The pattern consists of various bright and dark dots that are distributed in a non-correlating manner. The projected pattern is distorted by the geometry of the object. The depth is determined in several steps and for each pixel. First, the horizontal displacement of the dot pattern is determined based on the object distance. Based on the distortion, the depth of the respective pixel is then calculated in the next step using the equation for stereo triangulation [37]. For this purpose, the distortion in the unit of pixels, the base length (distance projector–camera) and the focal length in pixels are used. For each pixel, the distortion is determined using a local, e.g., 9×9 pixel area, which is compared with a set of reference images for different depths. The comparison is performed using cross-correlation. An interpolation is performed between the highest

correlation values to increase up to sub-pixel resolution [36,38]. For further information on the SL method using the *Microsoft Kinect V1* as an example, the reader is referred to [39].

Investigations of the *Microsoft Kinect V1* show the precision expressed by the standard deviation (SD) of 1 mm at 800 mm distance and of 11 mm at a distance of 3000 mm [32]. According to [31], the correctness (offset to the target geometry) is up to 40 mm for a captured distance of 1600 mm (within a typical working range of 400 to 4000 mm). Effects such as flying pixels (erroneous point measurement in a gap), color-dependent accuracy changes and multipath overlaps at edges do not or only occur at a very low level [31]. Moreover, for the *Matterport Pro 3D Camera*, which was used for online available training datasets by [40,41], the correctness, precision and resolution have been investigated in different studies. Here, a distance-dependent correctness of up to 80 mm for the furthest capturing distance was also determined. After a scaling factor is eliminated, a precision of better than 10 mm SD can be determined for the entire working range [35]. A LIDAR point cloud was used as a reference for the mentioned study. The resolution of the *Matterport Pro 3D Camera* is 5 (horizontal) and 10 (vertical) points per degree [42].

The ToF technique is based on measuring the travel time of a signal from an emitter to reflect at an object's surface and back to a receiver [30]. Pulse modulation (PM) and continuous-wave (CW) amplitude modulation are the most common ToF methods. In most depth imaging cameras, such as the *Microsoft Kinect V2*, CW amplitude modulation is used. In CW amplitude modulation, the object to be captured is continuously illuminated with NIR light, whose amplitude changes periodically. Because the signal needs a certain time between sensor and object, a phase shift occurs between the transmitted and received signal. This phase shift is proportional to the signal propagation time. If this time is multiplied with the known speed of light, the double distance between object and sensor system can be determined. The phase difference is determined for several modulated frequencies by correlating the received signal with the emitted reference frequencies. As long as the maximum distance is smaller than 2π of the frequency, a distance can be determined as unique [36].

The precision of the *Microsoft Kinect V2*, as with the *Microsoft Kinect V1*, depends on the acquisition distance and varies between 1 and 3 mm SD for the typical working range of 800 to 3000 mm [31,32]. Recent depth imaging cameras, such as the *Microsoft Azure Kinect*, have a precision of less than 1 mm for the same working range (static recording). Ref. [31] observed a constant offset of -18 mm for the whole working range of the *Microsoft Kinect V2*. Systematic erroneous measurements, such as flying pixels, color-dependent accuracy changes of up to 4 mm, multipath-effects at edges of up to 30 mm and a high dependence of distance measurements on temperature changes, are the disadvantages of this measurement principle [31,36,43]. These effects can be considered or eliminated in a later semantic segmentation.

LIDAR systems are used for static and kinematic recordings of scenes. LIDAR systems emit a laser beam, which is projected onto a rotating mirror. Through the rotation, the beam is shifted by a certain increment. For each increment, the vertical and horizontal directions as well as the distance to the surface are registered. Together with the intensity value, and eventually with further spectral values, the 3D point cloud is created. For the distance measurements, there is the phase difference (PD) method, which can be used to realize a higher measuring frequency, and the PM method, which is less object surface-sensitive [30]. PM LIDAR systems are preferred for kinematic scanning on mobile platforms. Kinematic laser scanning usually involves measuring individual profiles, which are assembled as an entire point cloud using navigation data or algorithms, as in [44]. Mobile LIDAR systems are mainly used for outdoor applications and on robots. Medium-range LIDAR systems such as *Velodyne HDL-64E* are often used for creating datasets in research projects with a precision of 20 mm [45]. High-end mobile mapping systems (MMS), such as the *Riegl VMY-1* [46], allow the surveying of large-scale areas with a point accuracy of 15 mm at 50 m distance and a precision of 10 mm. MMS such as the *Nav Vis M6* are used in many studies [47].

The current state of the technology for indoor surveys includes terrestrial LIDAR systems (TLS), such as the *Leica RTC 360*, *Z+F-Imager 5016* or *Faro Fokus X 3D 330*. These systems predominantly use the PD method and are used for distances shorter than 100 m. Laboratory and field investigations show that, with these measuring systems, 3D point clouds with precision of less than 1 mm and correctness of less than 2 mm in the near field of up to 20 m can be reached [48]. However, these values refer to optimal study circumstances such as matt or homogeneous surfaces. In practice, it has been shown for all LIDAR systems that the accuracy of the point clouds varies and scanning artifacts occur. Typical scanning artifacts are comet tails, mixed pixels on edges and multi-path effects on highly reflective surfaces, as shown in Figure 1. Other influencing variables, such as the measurement object, the setup and the environment, as well as the condition of the measurement systems [49], must be taken into account for the determination of the quality of a captured point cloud [50–52]. The resolution, the approximated accuracy, the acquisition method and the working range are crucial parameters that must be known or estimated for the later semantic segmentation of a point cloud.

2.3. 3D Point Cloud Datasets

In various reviews [23–25] and in web databases (e.g., <https://paperswithcode.com/datasets> accessed on 30 November 2021 and <https://www.semanticscholar.org/> on 30 November 2021) on point cloud datasets and methods for point cloud processing, an overview of more than 100 publicly available point cloud datasets is given. These contributions summarize information on application areas, applied sensors, environmental circumstances or file formats. The main goal of these publications is to provide benchmarks for arithmetic evaluations. A semantic segmentation is not available for all existing datasets. A selection of semantic 3D point clouds is examined in more detail. The focus will be on the initial human segmentation and its evaluation. Not all datasets could be documented in the same level of detail.

The datasets in Table 1 were derived from synthetic surface models. All show one object of one known class. In some datasets, the object models are subdivided so that they can be used for semantic and instance segmentation. Since the point clouds are derived from synthetic models, the geometry can be considered free of scanning artifacts. However, errors can still occur during annotation and alignment.

Table 1. Synthetic datasets with year of publication, data source, separation method (classification (Cls), semantic segmentation (SSeg) and instance segmentation (ISeg)), number of models, number of classes and environment.

Dataset	Year	Data Source	Separation Method	No. of Models	No. of Classes	Environment
ShapeNet [27]	2015	Trimble 3D Wareh., Yobi3D	Cls, SSeg	>220,000	3135	In-/Outdoor
ModelNet [53]	2015	Trimble 3D Wareh., Yobi3D	Cls, ISeg	151,128	660	In-/Outdoor
Shape2Motion [26]	2019	ShapeNet, Trimble 3D Wareh.	ISeg Cls, SSeg	2440	45	In-/Outdoor

An evaluation metric for classifications is introduced by the *ShapeNet* dataset, which describes how accurate or unique a classification is. Human annotators classify a semantic model until the classification accuracy varies by less than 2% [27]. The *ModelNet* dataset consists of 3D CAD models taken from web databases. The annotation is performed using *Amerzone Mechanical Turk* (AMT). The annotators classify different models using a web-based tool. For this, a model and a label are proposed. The annotators improve the correctness of a label for a displayed model by yes-or-no questions. An evaluation is conducted by the dataset designers for the ten most popular categories [53]. In the *Shape2Motion* dataset, a semantic segmentation of movable parts, such as wheels or car

doors, and their properties is performed. An evaluation of the classification is carried out by simulating the motion directly after the segmentation and classification [26].

Complex point cloud simulation tools, such as the *HELIOS++* [54] or *Gazebo* together with the *Robotics Operation System* [55], have reached a high level of development. These tools can be used to create point clouds from surface and CAD models that contain the characteristics of specific sensors and system configurations.

Indoor datasets are commonly captured with depth imaging cameras. Some of the most popular datasets are summarized in Table 2. For a large number of datasets, depth imaging cameras are used in combination with an initial measurement unit (IMU). Together with the poses from the IMU and the images, a Simultaneous Localization and Mapping (SLAM) procedure is used to compute a multi-dimensional representation of the captured scene. The semantic annotation occurs either in images, videos, meshes or in 3D point clouds.

Table 2. Indoor datasets recorded by depth cameras with year of publication, sensor, sensor method, separation method (classification (Cls), object detection (ObjD) and semantic segmentation (SSeg)), surface area and number of classes.

Dataset	Year	Sensor	Sensor Method	Separation Method	Surface Area Points	No. of Classes
SceneNN [56]	2016	Kinect v2	ToF	Cls, SSeg	7078 m ² 1,450,748	19
S3DIS [40]	2016	Matterport	SL	Cls, SSeg	6020 m ²	12
ScanNet [57]	2017	Occipial (iPad)	SL	ObjD, SSeg	78,595 m ²	17
Matterport3D [41]	2017	Matterport	SL	Cls, SSeg	219,399 m ²	40
ScanObjectNN [58]	2019	SceneNN, ScanNet	ToF, SL	Cls, SSeg	2.971.648	15

The *Stanford Large-Scale 3D Indoor Spaces* (S3DIS) dataset is semantically segmented as a 3D point cloud using the software *Cloud Compare* (CC) [59]. For the *SceneNN*, *ScanNet* and *Matterport3D* datasets, a mesh is the segmentation base. All annotations are performed with custom tools. The *SceneNN* dataset is first automatically segmented coarsely and then finely. The graph-based segmentation algorithm of [60] is adapted and the segmentation is afterwards improved by the operator by separating, merging and re-forming the segments. The semantic annotation is performed by users attaching labels to the segments [56,61]. The semantic segmentation of the *ScanNet* dataset is performed by automatic pre-segmentation and a subsequent fine segmentation with classification using tools on AMT. In addition to semantic segmentation with meshes, CAD models are fitted into a mesh and are available as a different data format [57]. The *Matterport3D* dataset is semantically segmented in two stages and verified by ten experts. In the first stage, floor plans are derived using planes projected onto the mesh. In the second stage, the meshes of individual rooms resp. regions are segmented according to classes and instances using *ScanNet's* tool [41]. For the *ScanObjectNN* dataset, the *SceneNN* and *ScanNet* meshes are the basis. A selection from this dataset is used and improved. Segments are rebuilt and categories are harmonized. A 3D point cloud with 1024 points is calculated out of each mesh.

The verification of depth image datasets is mainly performed by experts or the authors [41,58]. Alternatively, the same dataset is semantically segmented by different people to identify error annotations [56]. No information is available about the validation of the S3DIS dataset [40].

A selection of recent semantic 3D point clouds generated with LIDAR systems is summarized in Table 3. These datasets will be used later in the quality model. Most 3D point clouds from LIDAR systems are for outdoor scenes and are captured with multi-sensor systems (MSS). With MSS, the capturing of larger areas is more efficient than with TLS. The geometric accuracy of a few centimeters, which is necessary for the majority of applications in geodesy and civil engineering, is maintained. In addition to the

LIDAR measurements, many MSS capture RGB images from the scanned scene to colorize the point cloud. Furthermore, these images can be used for semantic segmentation.

The GT semantic segmentation of the datasets *Paris-Lille 3D*, *Semantic3D*, *MSL1 TUM City Campus* (MSL1 TUM CC), *Toronto3D* and *Complex Scene Point Cloud* (CSPC) is conducted completely or in parts with CC. For these datasets, the 3D point cloud format is the basis for data processing. This is also the case for the *SemanticKITTI* dataset, which is semantically segmented using a custom offline tool [13]. The *Building Indoor Point Cloud* (BIPC) dataset uses the *LabelMe* tool [62] for the segmentation and classification of 2D images. The 2D semantic segments are projected into 3D space after annotation. Any incorrect annotations in the point cloud are corrected using another 3D tool [63]. Another method to semantically segment 3D point clouds is to fit geometries, such as planes or boxes, into the point cloud. This is applied to parts of the dataset *Semantic3D* [15]. All points within a certain distance from the geometry are selected. The resulting segment is assigned to a class.

Table 3. LIDAR-recorded datasets with year of publication, sensor, sensor method, separation method (semantic segmentation (SSeg) and instance segmentation (ISeg)), number of points, number of classes and environment.

Dataset	Year	Sensor	Sensor Method	Separation Method	No. of Points	No. of Classes	Environment
Paris Lille 3D [64]	2018	Velod. HDL-32E	MMS car	SSeg	1431 M	50	Outdoor
Semantic3D [15]	2017	Unknown TLS	TLS	SSeg	4 B	8	Outdoor
SemanticKITTI [13]	2019	Velod. HDL-64E	MMS car	SSeg	4.5 B	28	Outdoor
MSL1 TUM CC [14]	2020	Velod. HDL-64E	MMS car	SSeg, ISeg	1.7M	8	Outdoor
Toronto3D [65]	2020	Teled. Opt. Mev.	MMS car	SSeg	78.3 M	8	Outdoor
CSPC-Dataset [66]	2020	Velod. VLP-16	MMS backp.	SSeg	68.3 M	6	Outdoor
BIPC-Dataset [63]	2021	Velod. VLP-16	MMS backp.	SSeg	-	30	Indoor

Closely related to the semantic segmentation is its evaluation. The *Semantic3D* dataset is evaluated by class comparisons in the overlapping areas of the neighboring point clouds. For this purpose, all points in the neighborhood of an adjacent point cloud are selected from a given point with a search radius of 50 mm. The classes of the selected points are compared with the class of the initial point [15]. The *SemanticKITTI* and the *CSPC* datasets are evaluated and improved by experts in a second processing step [13,66]. For the *BIPC* dataset, the segments created in 2D are evaluated on the 3D point cloud [63]. Statistical evaluation of semantic accuracy for all datasets is not documented. No information on the verification of semantic segmentation is available for the *Paris-Lille 3D*, *MSL1 TUM CC* and *Toronto3D* datasets.

Based on the datasets from the last six years, it can be concluded that more and more LIDAR systems are being used. Mainly LIDAR datasets of outdoor areas are created, because of the larger range and the higher resolution of these systems. For indoors, depth imaging cameras are still commonly used. Since many of these data come from the CV domain, surface models or voxels are additional output formats, along with point clouds and images. It can be seen that the datasets are not necessarily larger in terms of classes and points, but the annotation is more specialized and improved compared to early datasets. Earlier datasets are evaluated with new tools and optimized for specific tasks. The manual annotation can be still identified as a bottleneck.

2.4. Point Cloud Annotation Tools

Many annotation services and tools are used for autonomous driving or driver assistance. For this application, a few outdoor classes need to be (roughly) annotated. An overview and comparison of 33 annotation tools for this area of application is presented in [67]. These annotation tools mainly use simple geometries, such as bounding boxes,

plane and lines, to form instances and semantic classes. The very efficient and coarse semantic segmentation of large datasets is possible with these (semi-automatic) methods.

A number of commercial label services, such as *Playment* (<https://playment.io/> accessed on 20 November 2021), *scale.ai* (<https://scale.com/> accessed on 20 November 2021) and *basic.ai* (<https://www.basic.ai/> accessed on 20 November 2021), have also extended their services towards 3D point clouds. The disadvantage of these services is that they cannot be used for projects with confidential data. For applications where confidentiality and accurate semantic segmentation are relevant, offline tools can be used. Some of these tools are highly specialized for certain fields of application, so that only certain data can be imported or annotated according to predefined classes or rules [13,68].

The tools for annotation are diverse in terms of user interaction. In this context, annotation tools use virtual reality visualization [69]. Other tools use segmentation in 2D or 3D space [62,68], as well as fully manual and semi-automatic segmentation [70]. A selection of tools is briefly presented in Table 4. The tools are distinguished by the functionality of semantic and instance segmentation. In addition, they are categorized according to the central functions for the segmentation.

Table 4. Selection of annotation tools (x = present). Distinguished for instance and for semantic segmentation. Segmentation is performed in 2D or 3D with free-hand tools, automatically or with bounding boxes or geometries.

Tool	Instance	Semantic	Free Hand	Automatic	Bounding Box
Recap [71]		x	3D		3D
CloudCompare [59]	x	x	3D		3D
SemanticKITTI [13]	x	x	3D		
PCCT [72]		x		2D	

Recap [71] is a commercial software used to segment and classify 3D point clouds. Single or multiple registered point clouds are visualized in one project as one 3D point cloud. By rotations, displacements and zooming via mouse buttons, any perspective can be selected. In each *Recap* project, individual classes can be created, according to which a point cloud is semantically classified. For each class, an individual file is exported, which carries the class name. For the classification, point cloud segments are formed by free-hand selection, wrapping with simple geometries or fitting of layers.

CloudCompare [59] is one of the most commonly used open-source tools for point cloud processing and analysis, which can be used to segment and classify point clouds. Different methods to create annotations are offered as plugins. Moreover, *Semantic3D* and *MSL1 TUM CC* use the available functions in the main program for efficient semantic segmentation. The point cloud is displayed in 3D and navigation with mouse buttons is possible. The workflow uses the geometric and spectral point features to segment the point cloud in stages. After pre-segmentation, point cloud segments are further subdivided by free-hand selection. Individual segments can be combined into one semantic class, which is exported as a single file.

The **SemanticKITTI** annotation tool [13] was initially developed for the classification of kinematic 3D point clouds from a *Velodyne* LIDAR system. In addition to the point clouds, navigation and synchronization data are required for this tool. The processing of all *Velodyne* raw data is performed by this tool. Individual scans are registered at the beginning, resulting in a continuous acquisition sequence. For segmentation and classification, the point cloud is divided into 100×100 m tiles. The segmentation is carried out with a free-hand lasso and a point marking tool. Predefined or custom classes can be used. The original point cloud files are not modified with this tool. For each point, a label file is created that contains the semantic information and instances of each point.

The **point cloud classification tool** (PCCT) [72] is a tool for the semantic segmentation of (primarily) static panoramic scans. Point clouds are projected into 2D space for classification. This is achieved by cutting the point cloud horizontally or vertically into

slices. Alternatively, vertical cylinders in different distances are used as a projection plan. The segmentation is performed in the 2D plane using a pixel-based regional growing method [60]. Via a browser application, users can assign one of 20 predefined indoor and outdoor classes to the displayed segments. The PCCT is multi-user-capable. For each semantic class, a point cloud is provided.

The presented tools show the range of functions for the segmentation of the point clouds. As more flexibility is given to the user for segmentation, more details of the segments can be formed. Tools such as the PCCT form the segments according to fixed rules. Here, different results can only be achieved by the classification of different users. With all other tools, classification and segmentation performance are not separable.

3. Quality Model for Semantic Point Clouds

Many published datasets and tools are indicated as high-quality. This statement is true for the application for which these datasets are intended. The term “high-end” may refer to the quality of the acquisition, the reconstruction of a mesh, the semantic segmentation or any other aspect. However, in the rarest cases, all possible aspects are of high quality. In order to describe the quality of the datasets, the first step is to define the main quality characteristics. Unfortunately, a quality model cannot be created for all conceivable cases. This would be too complex and no longer understandable, and the focus should therefore be on one aspect per quality model. This aspect will be the focus of semantic segmentation for our quality model. The measurement methods, datasets and annotation tools described in Section 2 are the basis for the quality model’s development.

One approach to describe the quality is to use the ISO 9000:2015 (3.6.2) [73] and DIN 55350:2020 [74]. Here, quality is defined as the “degree to which a set of inherent characteristics of an object fulfills requirements” [73]. The point cloud and the segmentation processes are the subjects of investigation, whose quality characteristics should be fulfilled to a certain degree. The characteristics can be expressed by quality parameters. Thus, the quality is a simple comparison of the actual and required quality parameter values of an object. ISO 9000:2015 also defines quality for the process of creation, so that process characteristics are required as well. Besides the process (segmentation) characteristic, there are characteristics that are affected by previous steps, such as capturing, and this influences the final object characteristics (semantic point cloud). This interaction is shown in Figure 3. The prior characteristics are derived from the acquisition method and must meet a minimum standard. Only if the minimum standard is fulfilled can the actual processing step be performed. The unprocessed point cloud must have a minimum resolution and fulfill a certain geometric *level of accuracy* (LoA). A suitable scheme to define the geometric LoA is provided by DIN18710:2010 [75]. The prior and the process characteristics influence the new object’s characteristics.

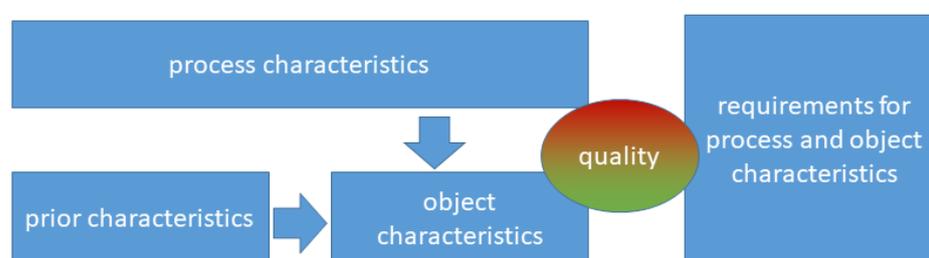


Figure 3. Interaction of the different characteristic types and the requirement to determine quality.

The way in which a semantic segmentation is performed can be expressed in the object’s quality parameters. For example, erroneous points should be determined by the semantic knowledge of the annotator and should be assigned to an appropriate class. Thus, additional knowledge is introduced into the application. The quality of this knowledge is an example of a process-dependent influence on the quality. Aspects such as how a process

is carried out and how it is evaluated must be expressed by characteristics. For the semantic segmentation, the degree of correctness and repetition accuracy can be determined if either a reference is available (correctness) or the process is performed n times independently (repetition accuracy). For the repetition accuracy, the number of repetitions and the type of process must be defined. This is an additional aspect that should be covered by a quality model. For the creation of a quality model that takes into account the above-mentioned aspects, the following basic requirements are necessary:

- An application must be defined;
- A semantic segmentation process must be described;
- An abstract model of the semantic must be created;
- A data model must be created;
- Measured or synthetic point clouds must be available;
- Characteristics and parameters must be defined;
- Target values for the quality parameters must be defined.

These seven constraints set the framework for the development of the quality model. The applied case is the creation of a semantic segmented point cloud for the modeling of indoor components and furniture. One application for which such a scenario is necessary is the creation of a Building Information Model (BIM) from a point cloud (Scan-to-BIM). The semantic segmentation of a point cloud for this is a complex and an increasingly demanding application in geodesy and civil engineering [76]. The differentiation of clutter or scanning artifacts from filigree classes, such as tables or chairs, is a problem that cannot be adequately solved by current automatic processes.

3.1. Classification Process

A process description outlines the individual steps that are to be implemented. Thereby, goals (tasks), data, definitions, tools and framework conditions are addressed. Point clouds belong to the group of geodata, so that a process description based on the model for geodata [77] is chosen. The point cloud semantic segmentation process is shown in Figure 4. It consists of the object in the real world, two models (c and d), the data (b) and an action statement describing the interaction.

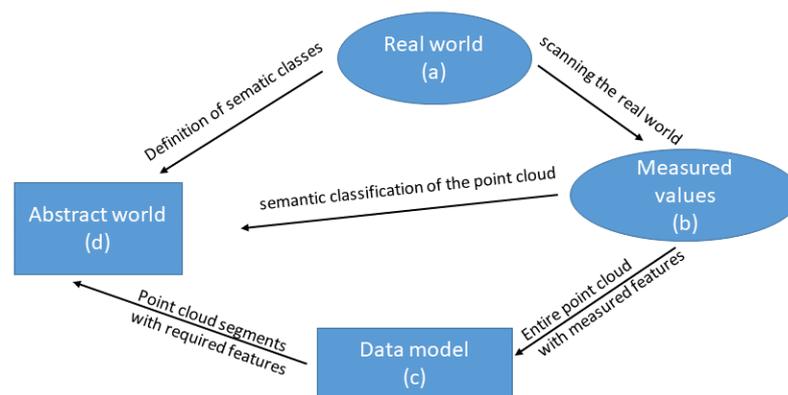


Figure 4. Process for semantic point cloud segmentation.

Semantic segmentation is an extension of point cloud features that can be described as a process. An abstract model (d) of the reality defines semantic classes, which describe which objects are represented and in which level of detail. The abstract model is always a generalization of the real world (a), which is captured by measurement methods as measured values (b). The measured values are the unclassified point clouds or individual points and consist of the geometric and spectral features. The data model (c) defines the file format in which the measured values are available and into which format they are transformed for the abstract model (by semantic segmentation). The data model for

semantic point cloud segmentation specifies that any point has a new semantic feature and each semantic class is a segment.

3.2. Abstract Model

The abstract model for a semantic segmentation describes which semantic classes are represented and defines the class content. The definition of the abstract model should correspond to the application for which the point cloud is used. When defining the classes, two variants are used. Variant 1 is to determine exactly those classes that are needed. Variant 2 is a hierarchical class definition (CD). For this, super-classes are formed stepwise, so that object parts can be distinguished. Variant 1 leads to very small semantic CD, such as that required for autonomous driving [78]. Variant 2 leads to a CD with more than 50 classes [64]. A very small number of classes has the advantage that the semantic segmentation can be performed faster and the classes can be more precisely defined. A distinction between trees and traffic signs is easy in point clouds. If it is necessary to distinguish between beech and oak trees, the definition is much more complex. It should be done in multiple steps and with additional training of human or algorithmic annotators. In such a case, the definition of the abstract model should be structured hierarchically, as shown in Figure 5. A simple structure in two stages is applied to the *SemanticKITTI* dataset. There, the first hierarchical level contains the class soil, which is distinguished in the second level by roads, sidewalks, parking lots and other surfaces [13]. This structure makes the semantic segmentation more explicit and simpler.

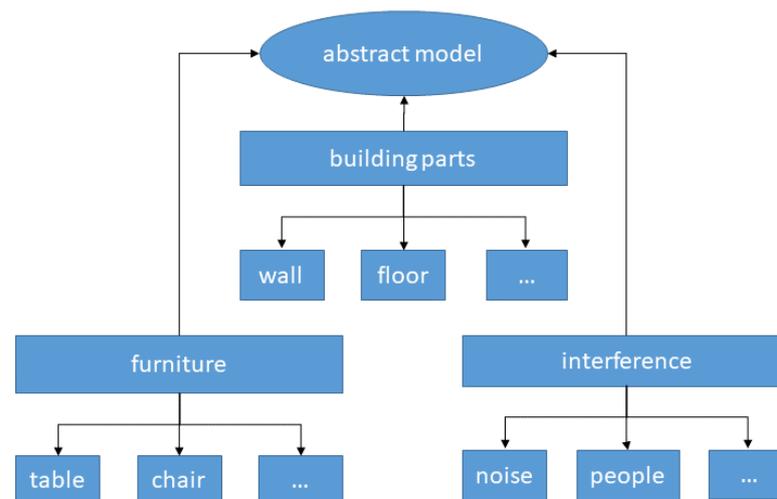


Figure 5. Hierarchical abstract model for the definition of semantic classes. The application is the classification of indoor point clouds.

The optimal abstract model contains all possible classes. This is not possible due to the wide range of applications for point clouds. In Figure 5, a two-level model is shown, where only classes (or objects) are considered that are in a building. It would be too specific for modeling an entire building, since no external objects are included. In turn, for modeling parts of a building, this model is too general, because, in such a model, furniture and disturbances are not included. The advantage of a detailed model is that classes that are not needed are simply ignored. This favors a general model. A possible way to build a universal model would be to refer to the linguistic model *WordNet* [79], as it is already the basis for *ShapeNet* [27] as well as others. All nouns are attributed to the word entity. Starting from the entity, top-level nouns are formed, which can be distinguished in any direction. An application-independent hierarchic abstract model can thus be built. *WordNet* also has the advantage that cross-connections between hierarchical classes are possible and it has a directing effect for the creation of specific abstract models.

For building models from point clouds, an orientation to existing standards, such as the Industry Foundation Classes (IFC) [80], as well as national [81,82] and international [83] guidelines for the Level of Development (LoD), would be possible and helpful. Unfortunately, these standards and guidelines do not yet offer an exact and detailed description of what a semantic class has to look like, but they regulate in which level which contents have to be presented. Moreover, it is still necessary to create an explicit CD. This is the basis for the work of the annotators. The following points should be considered when defining the abstract model:

- A general semantic model should provide the structure for the abstract model;
- The classes of the abstract model should be structured hierarchically, so that, in one definition level, only a number of around five classes exists. The next lower-definition level should contain only points of one higher-level class;
- For each level, all points are classified;
- The classification is an iterative process;
- The level of detail is mainly based on the application and the existing technical functions of the tool.

In addition to class names and class structures, the content of the classes must be defined and represented in such a way that it is understood by the annotator without doubt. The following points for the content definition should be considered:

- The semantic definition must be written in the language of the annotator in order to avoid linguistic misunderstandings, such as translation errors.
- Objects must be described unambiguously by describing their shape, size or color. It is to be considered that objects of the same semantic class are represented differently in the point clouds. If objects appear in different designs, then this is to be described adequately. A definition of objects can be created, as described in [84].
- Special and unknown objects are to be illustrated by examples, so that the idea of the annotator is identical with what is being represented.
- A definition consists of a written and a figurative description.
- Topological relations should be represented to facilitate the decision in case of difficult-to-recognize object appearances. For example, furniture could be defined as standing on the floor or walls running perpendicular to the floor.
- Geometric boundaries should be clearly defined, as this is the only way to achieve the required geometric accuracy. Using the class *door* as an example, the following definition is possible: *A door ends at the frame, at the seal or at the wall. Erroneous points should be separated completely from the objects.*

The abstract model setup must be communicated to the annotators in a suitable format (training) and checked on a regular basis. In [13], this is implemented by training the annotators on the data previously and providing feedback on the performance. This is possible since each point is semantically segmented by at least two annotators. Feedback to annotators can be given directly by moving or highlighting the segment in the tool [26]. In addition, videos, teaching tools and abstract models are common and useful [67].

3.3. Data Model

The data model is defined differently in the literature. Occasionally, the *abstract model* is also called the *data model*. In the context of this paper, the definition by [77] is applied, who sets the data model equal to the *physical model*. The data model defines which file format is to be used for the measured and the semantic point cloud. In addition, it is clarified how the objects are organized in this file format and which attributes an object can have.

The data model consists of two layers. One is the unclassified point cloud and the other is the classified point cloud. For the purely semantic segmentation of single point clouds, a very simple data model can be chosen. It provides the point cloud as an unsorted list of points with their geometric and spectral attributes. For the unclassified point cloud,

these are typically 3D positions (x , y and z coordinates), color values as RGB values and reflected intensity as values. If more attributes are needed for the point feature description, many applications use a database. Besides the structure and file format, the data type of the feature has to be defined. This is usually done in the file format description. A data model for the semantic segmentation must be able to represent semantic features in addition to geometric and spectral features.

3.4. Quality Model

The quality model describes the characteristics and suitability of a semantic segmented point cloud for a certain application. Process- and object-specific quality characteristics from the quality domains of reliability, integrity (usefulness) and accuracy are used as the basis for the evaluation [85]. The quality characteristics are chosen in such a way that they are applicable for manual semantic segmentations, if these are performed according to the process in Section 3.1. The three quality areas are described by seven quality characteristics (Figure 6). Each quality characteristic is expressed by quality parameters.

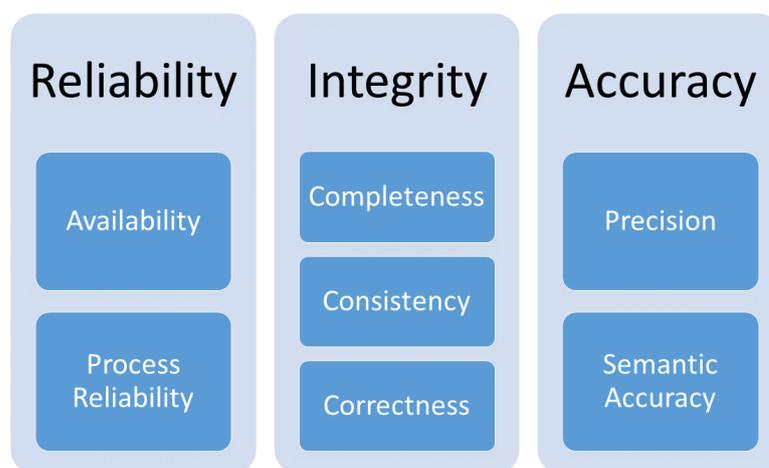


Figure 6. Quality model for 3D point clouds. Concept idea inspired by [85].

The model shown in Figure 6, which is further described below, has its roots in the idea of [85]. It is accepted in many disciplines and is used for various applications. To the best of our knowledge, this model idea has not yet been applied to the semantic segmentation of 3D point clouds. Our central contribution in terms of the quality model is the compilation and selection of characteristics and parameters to make a semantic segmentation of a point cloud describable and evaluable. The development of the quality model has the goal of questioning and improving the quality of the datasets. Only with all-round high-quality datasets is it possible to develop reliable and accurate algorithms and tools [86]. In addition, the practical use of point clouds for reality capturing should be considered by the model characteristics.

3.4.1. Quality Characteristics

Quality characteristics are selected characteristics from the total of all characteristics that a semantic point cloud has. This characteristic selection relates to the requirements of a certain application in which the object is to be used [74]. When applying a quality model to a semantically segmented point cloud, the quality of the segmentation and classification, as well as the quality of the raw point cloud, is primarily expressed by the quality characteristics.

The **availability** describes which data and information are available to the annotator or the algorithm prior to the task. Parameters for this characteristic express which information is known about the point cloud. This relates to point clouds, task definitions and processes. A quality parameter expresses whether the information is available in a specific form and

in the required quantity. It is the basis of all further characteristics and must be fulfilled in order to carry out a semantic segmentation and its evaluation.

The **process reliability** describes how the process was carried out. This characteristic can be determined by performing the semantic segmentation several times. After each segmentation, accuracy parameters are determined, which can be used as terminating criteria, as in [27]. Other variants require that a certain number of iterations must be fulfilled in order to determine a quality parameter. This variant is preferred for the description of the process quality, since it maps the variance of the metric and parameter values. Using this, the achievable performance can be determined by a process setting. Due to the complexity of these tasks, the average repetition factor to determine this parameter is usually very small, as shown in [13] factor 2, in [57] factor 2.3 and in [68] factor 4.

Completeness gives the degree to which the necessary information, determinations and execution of the work steps for the classification are present.

Consistency is the degree to which the measured values match the data model. Here, it is necessary to check whether the point cloud features are present and whether they take the corresponding range of values.

For the semantic segmentation, **correctness**, **precision**, and **semantic accuracy** are different characteristics that have different underlying causes and different effects on the usability of the point cloud. Moreover, these characteristics are often defined and summarized in different ways. For example, if only the performance of a semantic segmentation method is to be considered, correctness and precision are often combined with accuracy. The accuracy is commonly given when ML and AI algorithms are used. In many semantic segmentation applications, this defined characteristic is expressed by the parameter IoU, also known as the Jaccard index, or the F1 score, also known as Dice's index. These parameters are the weighted averages of both characteristics. It is advantageous to apply one combined characteristic of accuracy and its meaningful parameter, e.g., IoU, for better comparison. Other applications use more than one parameter to describe the different perspectives of accuracy for a more distinguished and cause-oriented view.

For the analysis of the semantic segmentation process, two types of errors are possible: a point is erroneously assigned to a class to which it does not belong or a true point of this class is not recognized as member of it. These two errors are known as first- and second-type errors from statistical tests [87]. The segmentation **precision** can be considered an error of the first type. This error specifies how well an annotator or an algorithm can distinguish classes—for example, how accurately class boundaries can be drawn. The segmentation **correctness** can also be considered a second-type error. This error describes how well a class can be recognized, e.g., how unique the point features are. Thus, the best features are used to obtain a class of homogenous points. This type of error can be of importance depending on the analysis in question. For example, it may be less critical if not all points of a large class (such as floor) are detected during semantic segmentation, as long as these points are not classified or assigned to a class (e.g., scanning artifacts) that is not further used. More problematic are additional points (e.g., from scanning artifacts) that are assigned to the class floor, because the point cloud represents incorrect semantics.

Thus far, correctness and precision based on the number of points describe the quality of a semantic point cloud. However, these characteristics do not give any information about the geometry of the semantic classes and its geometric size changes due to errors. In order to be able to evaluate the geometric aspect as well, the characteristics of correctness and precision have to be extended. Geometric correctness can be determined if a (dense) reference point cloud or surface model is available. The correctness can be determined for each individual point. This information can no longer be evaluated for several hundred thousand points. The correctness of a point cloud can be determined by the mean, average deviation or standard deviation of all points in a segment. In general, correctness is the degree to which the abstract model matches the achieved semantic segmentation result. This can be divided into user-dependent and software-dependent correctness. The user-dependent correctness is based on the understanding of the CD and the usage of the

software by the user. The software-dependent correctness refers to errors in the software (e.g., incorrect parameters or programming). However, a separation is only possible if the semantic segmentation is carried out several times under controllable conditions.

The quality characteristic of precision is described by the parameters for the semantic and geometric precision. The term "precision" should be defined clearly, because there are different definitions in use. In the geodetic context, precision is often understood as repeatability [87]. The deviation of the results of an experiment to its mean value after n repetitions is determined. For the semantic segmentation process, this definition would lead to the determination of how much the individual segmentations deviate from each other. This shall not be the main subject of the investigation, since a deviation to the mean of several segmentations usually has no relevance for a practical application. Nevertheless, it makes sense to repeat a segmentation and to calculate a joint point cloud from these repetitions in order to increase the reliability, as mentioned above. Usually, deviation from a reference point cloud is required. This can be expressed by the ratio of true points to all points assigned to a class [88]. This term describes how much of the segmentation is "correct" and is commonly used in ML. Mostly, the inverse proportion is of major importance for the development of an application, because this describes what does not work yet [89]. This proportion is then the subject of analysis. In addition to the use of the number of points, it is advantageous for 3D models and point clouds to also use the areal ratios as well as geometric parameters.

The **semantic accuracy** describes how well the semantic label fits to a semantic point cloud segment. The difficulty is in defining what is semantically correct, which attributes are described and which depth of description and distinction must be applied. For the definition of what is semantically correct, no universally valid definition can be found. An attempt to standardize this problem was discussed in Section 3.2. For the type of attribute description, the IFC standard [80] can be used. This is designed for the development and not for the documentation. This can be explained using the example with the tables. The table itself forms a semantic class. These classes can be differentiated during the next stage into a frame and table top. As far as we know, there is no standardized scheme for this definition, so that an individual CD as shown in Appendix A must be developed and applied.

3.4.2. Quality Parameters

The seven quality characteristics used for semantic segmentation can be described by quality parameters. These parameters describe the property that an object has for a certain characteristics. For instance, these parameters are the presence of a certain data format as a qualitative parameter or the *number of points* (NoP) as a quantitative parameter. This will be demonstrated in an example in Section 4.1. The evaluation of point clouds by the quality model will be covered in Section 4.2. For the evaluation, the quality parameters must be determined and threshold values must be set. Furthermore, the parameters for the semantic segmentation can be distinguished into parameters with object relation (O), concerning the point cloud, and process relation (P), such as the time required for an action or the use of a certain CD. All parameters for the semantic segmentation task are briefly explained and shown in Tables 5–11. The parameters are numbered in the text and refer to the corresponding table entry with P#.# for a clear understanding.

Quality parameters for characteristic **availability** describe which information must be available about the process and the point cloud for a description and an evaluation (Table 5). These parameters are the abstract model expressed by the CD (P1.1), the size of the point cloud expressed by the *NoP* (P1.2) and the *area size* (P1.3), as well as the object features (e.g., x -, y -, z -coordinates) before (P1.4) and after (P1.5) the semantic segmentation. Furthermore, the *file format output* (P1.6) and *use restrictions* (P1.7) must be investigated. The *use restrictions* refer to the question of whether a dataset can be used for an application or processing step. Further restrictions are that certain datasets may not be used for training.

The parameter P1.7 ensures an objective evaluation of the datasets. Thus, it is considered that any dataset has a certain bias, which is learned by ML algorithms [86].

Table 5. Parameters for availability.

P. No.	Parameter Name	Unit	Range	P/O
P1	Availability			
P1.1	CD exists		yes/no	P
P1.2	Number of points		>0	O
P1.3	Area size	m ²	>0	O
P1.4	Object charac. in		yes/no	O
P1.5	Object charac. out.		yes/no	O
P1.6	File format out		e.g., pts	O
P1.7	Use restriction		yes/no	O

Table 6. Parameters for process reliability.

P. No.	Parameter Name	Unit	Range	P/O
P2	Reliability of Process			
P2.1	Number of segmentations		>1	P
P2.2	Average time required	%	0–100	P

The parameters *number of segmentations* (NoS) (P2.1) and *average time required* (ATR) (P2.2) describe the **reliability of the process** (Table 6). If a point cloud is independently semantically segmented more than once, the reliability can be measured. The more frequently a process is carried out, the more reliable are the correctness and accuracy. This is the theoretical assumption. The parameter NoS describes how often a segmentation was performed with a certain method. It is the basis for the calculation of other parameters and can also be used as a quality measure. The ATR can be used to compare different semantic segmentation methods. The ATR is calculated for each method. The average time of all annotators with any method is of interest. The maximum segmentation time of all methods is the value Δt_{max} . The ATR is calculated from Equation (1), where i stands for the respective segmentation. Δt_i is therefore the time needed for the segmentation i . Moreover, the user-dependent segmentation time can be analyzed if all segmentations performed with a certain tool are compared. The parameter ATR describes the process and allows the planning of the working time.

$$ATR = \frac{\sum_{i=1}^{i_{max}} \|\frac{\Delta t_i * 100}{\Delta t_{max}}\|}{i} \quad (1)$$

Table 7. Parameters for completeness.

P. No.	Parameter Name	Unit	Range	P/O
P3	Completeness			
P3.1	Semantic segmentation rate	%	0–100	O
P3.2	Number of classes		>0	O

The **completeness** of a semantically segmented point cloud (Table 7) is described by the *semantic segmentation rate* (SSR) (P3.1) and *number of classes* (NoC) (P3.2). The parameter SSR describes how many points have been assigned to any class. The SSR is the quotient of the number of classified points (P_{cls}) and all points (P_{all}) (Equation (2)).

$$SSR = \frac{P_{cls}}{P_{all}} * 100 \quad (2)$$

A point cloud that is only segmented in parts often occurs in the application phase. The semantically segmented parts of the point cloud are used for training or for the evaluation

of an algorithm. The rest of the data are then semantically segmented using the automatic method. The parameter NoC describes how many classes are available for a certain dataset.

Table 8. Parameters for consistency.

P. No.	Parameter Name	Unit	Range	P/O
P4	Consistency			
P4.1	Geometric Consistency (GC) of x, y, z	m	≥ 0	O
P4.2	Spectral Consistency of RGB (SCRGB)		0–255	O
P4.3	Spectral Consistency of I (SCI)		0–255	O
P4.4	Class equality		0–1	O

The **consistency** of the data (Table 8) is determined by the units and the scaling ranges of the object features (P4.1 to P4.3). Each object parameter directly relates to a quality parameter. The determination can be achieved automatically or taken from the data (e.g., using a text editor). Furthermore, the consistency is described by the measure of the *class equality* (CE) (P4.4). This is calculated from the target value of a balanced class distribution (C_{target}). All classes should be represented by the same amount of points, so that, later, an ML procedure has optimal learning conditions. However, this requirement is never given with real datasets, because classes such as walls and floors are overrepresented by points. The proportion of points of a class in relation to the total NoP is expressed by a ratio in the value range 0–1. The actual distributions are then calculated (C_{act}). The differences between the target and actual values for each class are determined. The sum of the absolute differences divided by two is a measure of balance (Equation (3)), where 0 represents a balanced ratio and 1 an unbalanced ratio.

$$CE = \frac{\sum_{i=1}^k \|(C_{target} - C_{act})\|}{2} \quad (3)$$

Table 9. Parameters for correctness.

P. No.	Parameter	Unit	Range	P/O
P5	Correctness			
P5.1	Recall of points <i>class x</i>	%	0–100	O
P5.2	Recall of area <i>class x</i>	%	0–100	O

The **correctness** (Table 9) of the semantic segmentation can be described by the parameter *recall of points* (RP) (P5.1). The PR is the rate between correctly assigned true positive (TP) points and the NoP in the abstract model for a certain class (TP and false negative (FN) points) (Figure 7). It is expressed by Equation (4). This parameter depends on the size differences of the class in the abstract model. If the classes differ greatly, as can be evaluated by the parameter CE, a comparison of different classes may lose significance. For a small set, even a few FN points can significantly lower the parameter. This problem is discussed in [89] and described by a new parameter for informativeness. For applications in the context of point clouds, this parameter is unsuitable due to the irregular distribution of the points.

$$RP = \frac{TP}{TP + FN} \quad (4)$$

$$RA = \frac{TP_{area}}{TP_{area} + FN_{area}} \quad (5)$$

To avoid the point cloud density problem, the representation in the form of areas can be used. Here, the areas are calculated for the point cloud segments. Instead of the NoP, the TP area size can be inserted into Equation (4). The result is the *recall of area* (RA) in Equation (5). The correctness is now described by the area that is covered by TP points

divided by the area of all reference points of this class. As an intermediate step to calculate these parameters, the areas that are correctly and incorrectly assigned are calculated. In the case of incorrect assignments, the distinction between FN and FP areas is of interest. The parameter RA expresses the influence of FN surfaces. The influence of the false positive (FP) areas is described in the following, among others, by the *precision of area* (PA). The FN and FP points are visualized in Figure 8. This visualization allows an analysis of the semantic segmentation, e.g., the assignment of scanning artifacts to a class or the occurrence of classification gaps can be determined.

Result of the semantic segmentation

	Floor	Table	Chair	Scanning Artifacts
GT semantic segments	Floor	TP	FN	
	Table	FP		
	Chair	FN		
	Scanning Artifacts			

Figure 7. Schematic representation of the confusion matrix for the floor class with entries for TP, FN, FP and true negative (TN) points.

Table 10. Parameters for precision.

P. No.	Parameter	Unit	Range	P/O
P6	Precision			
P6.1	Precision <i>class x</i>	%	0–100	O
P6.2	Precision area <i>class x</i>	%	0–100	O
P6.3	MD of FP pts. <i>class x</i>	mm	≥0	O
P6.4	SD of FP pts. <i>class x</i>	mm	≥0	O

The precision is expressed by the *precision of points* (PP) (P6.1) and the PA (P6.2). The PP is the ratio of TP points of a class to all points assigned by the segmentation of this class (Equation (6)). The assigned points could also be expressed as the sum of the TP and the FN points (Figure 7).

$$PP = \frac{TP}{TP + FN} \quad (6)$$

$$PA = \frac{TP_{area}}{TP_{area} + FN_{area}} \quad (7)$$

The consideration of the characteristic precision based on areas that are spanned by the point cloud segments can be advantageous when using the point cloud as a model. For

a geometric expression, Equation (7) can be used to determine PA. The visualization of the FP points is given in Figure 8, which is a good starting point for the analysis process.

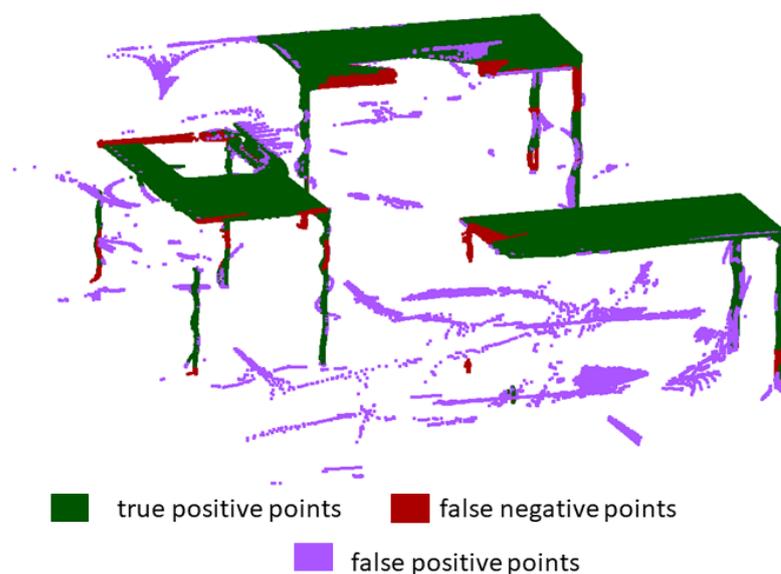


Figure 8. Segmented point cloud of the class table colored by TP, FP and FN points.

The geometric part of the precision can also be described by the parameters *maximum deviation (MD) of FP points* (P6.3) and *SD of FP points* (P6.4). The *MD of FP* and *SD of FP points* rely on the FP points of the semantic segmentation. They are the points that change the geometry of the semantic class, as shown in Figure 9. For this consideration, only classes with semantic objects are considered, since, normally, the goal of semantic segmentation is to extract objects and to remove scanning artifacts. The geometric deviation of the point cloud segment is of major importance for creating a model. If the point cloud is used to create a mesh, then the MD, which is the enlargement of the class segment, is decisive. This is expressed by the furthest FP point. For modeling on the basis of point clouds or the representation of the recorded objects by symbols, as is the case at the LoD 100 for a BIM application [90], the parameter *SD of FP points* is more meaningful.

The **semantic accuracy** (Table 11) is described by parameters that can be expressed by yes-or-no questions. Documentation of the process and visual inspections can be used to determine the *CD applied* parameter (P7.1) and whether it is structured hierarchically (P7.2). The parameter *CD applied* can be answered with *yes* if the CD is used and at least one class is segmented. The parameter *Hierarchical CD* can be confirmed if the used CD has several levels (at least two) and so different semantic detailing levels are available. The query whose class was finally used is expressed by the parameter P7.3. If the class is present and semantically correct, the parameter is answered with *yes*.

Table 11. Parameters for semantic accuracy.

P No.	Parameter	Unit	Range	P/O
P7	Semantic Accuracy			
P7.1	CD applied		yes/no	P
P7.2	Hierarchical CD		yes/no	O
P7.3	<i>class x</i> used		yes/no	O

3.4.3. Descriptive and Evaluative Function

A quality model such as the one above can have two functions. One is descriptive and the other is evaluative, as described by ISO 9000 (2015) [73].

For the **descriptive use**, the aim is to display and analyze how individual parameters (defined as significant by the model) vary when influences change. Different settings, tools or work processes for a semantic segmentation can be compared. Quality parameters are not transformed into another representation or range for this purpose. The main influencing characteristics for the development of a semantic segmentation process are considered and this is one main application of the quality model. More precisely, the influence of the initial (manual) segmentation of a point cloud is investigated. Thus, the model also provides the basis for describing an automatic (e.g., ML-based) semantic segmentation process, as considered in many works, such as [91–94].

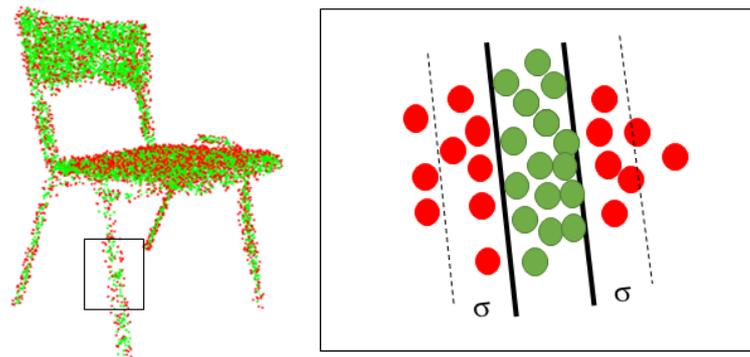


Figure 9. Calculation of the SD of FP points σ on the example of a chair. Green TP points are within the object boundaries. The red FP points were added to the chair class but actually belong to another class.

For the **evaluative use**, the suitability of a point cloud for an application should be assessed. It should be derived from the parameters whether a point cloud in combination with the segmentation method is suitable for a certain application or not. For this purpose, the calculated parameters of the quality model are crucial. An example application would be to use a semantic point cloud to determine the wall surface area, to calculate the renovation costs, based on the as-built wall surface area. For this task, correct semantic segmentation is crucial. The point cloud should be evaluated by applying a quality model in advance. The quality of the individual parameters must be defined by limit or target values. These values are derived from the application. The evaluation steps are defined according to the scheme shown in Figure 10.

After the limit or target values have been defined, they are compared with the determined actual values. This adjustment can be represented in an automatic procedure by one Boolean value. In the simplest overall evaluation method, all parameters must be true for sufficient quality. The weighting of the parameters for special cases prevents excessively rigorous filtering. The central issue is the limit or target values, which are not always known and have to be estimated based on experience.



Figure 10. Evaluation of the suitability of a point cloud with the quality model.

4. Applying the Quality Model

The benefit of the quality model as a basis for describing and evaluating the properties of a semantic 3D point cloud will now be explained by some examples. The performance of the quality model is shown on the basis of two of our own indoor point clouds and other publicly available point cloud datasets. Our own point clouds are shown in Figure 11 and were semantically segmented independently, multiple times, using two different semantic segmentation tools. The quality of the point cloud and the semantic segmentation process are described by the quality parameters. The evaluation performance of the quality model is considered for our own and the publicly available datasets. The applications of interest are the analysis of:

- Semantic point cloud as a model;
- Semantic point cloud as a modeling basis;
- Semantic point cloud as training data.

Target values are defined in each case. The geometric, semantic and formal characteristics of the point cloud are processed and used for an application. However, these point cloud characteristics have a degree of uncertainty if the semantic point cloud was created by capturing a real object and performing a semantic segmentation. The possible errors and the quantitative uncertainty of the sensors are described in Section 2.2. It can be stated that the usually resulting effects of currently available and used sensors do not significantly affect the indoor modeling applications. Our own point clouds were recorded with the *Z+F Imager 5016* using a resolution of 6 mm at 10 m. The quality was set to *high* to reduce the noise while still having a moderate (in practice useful) recording time of 6 min [95]. The geometric correctness of this point cloud on a flat surface can be estimated as 2 mm to 3 mm according to the investigation of [48], using the *DVW-test-field-method* according to [96]. This accuracy varies due to the different surface shapes and other object properties. In addition, scanning artifacts occur, as shown in Figure 1 and described in Section 2.2. The focus is now on the semantic segmentation, where errors are caused by tool settings and the annotator.



Figure 11. Points to be examined without semantic segmentation. Objects of the chair, table and floor classes, as well as scanning artifacts, are shown.

The point clouds in Figure 11 are very challenging for semantic segmentation. A CD was developed and applied in order to investigate segmentation problems. This CD consists of five classes and is partly hierarchically structured. The classes of the first level are floor, furniture and scanning artifacts. In the second level, the furniture class

is divided into table and chair. Tables and chairs are two object classes that are spatially and geometrically similar, which makes segmentation difficult. The points of these two classes also have similar spectral properties. Finally, the object surfaces are highly reflective and the geometric shape is susceptible to the occurrence of scanning artifacts. The floor class was integrated to simulate scenic segmentation with foreground and background objects. The separation of scanning artifacts is a complex task, even for humans, where subjective decisions must be made and learned. The test point cloud does not represent any real particular task, but is intended to demonstrate achievable performance on challenging cases. The point clouds show recordings of a laboratory (*Lab*) and a seminar room (*Room*), which were automatically segmented with the *PCCT* using the spectral parameters color and intensity. The point clouds were processed by up to nine different annotators. These are the test point clouds *Lab RGB*, *Lab I* as well as *Room RGB* and *Room I*. Furthermore, the point clouds *Lab* and *Room* were processed with *Recap*, in which the annotators determine the segments by themselves. These are the datasets *Lab R* and *Room R*.

4.1. Quality Model to Describe Semantic Point Clouds

The description of a semantic point cloud and a segmentation process is always based on a selection of characteristics, with the goal of being able to answer a specific research or practical question. The research question for the following consideration is:

What influence do the segmentation tool and different annotations have on the quality of the semantic segmented point cloud?

The motivation for this question is to develop an efficient, effective and traceable segmentation process. Different experimental settings and development stages shall be described, so that their influences on the process can be analyzed. This should also result in more convenient point clouds for models and training data, as well as improved process and algorithm understanding. All characteristics of the model are described in detail in the following.

4.1.1. Reliability Characteristics

The reliability of a point cloud can be described mainly by formal information or metadata, as listed in Table 12. The creation and use of a CD, which regulates which objects will be segmented and classified, is of primary importance. A comparison of semantic segmentation is only possible if the CD is kept constant. The parameter *CD exists* must be available to utilize all other semantic-based descriptions. The accuracy of the implementation of the CD is described by the parameters of semantic accuracy in Section 4.1.3. For the test point clouds, a CD exists, which describes the semantic classes of floor, furniture, chair and table, and scanning artifacts.

The size of the point cloud is another formal parameter, which is described by the NoP and the surface area. The NoP that can be processed by segmentation tools varies widely. Sometimes, the point cloud is automatically reduced to a maximum NoP. This filtering changes the point cloud structure and, depending on the application, can result in unwanted effects, such as the loss of surface details. The *Lab* and *Room* point clouds consist of 2.7 and 14.5 million points. The surface area of the objects covered by points is 51 m² and 61 m² for the *Lab* and the *Room* point clouds, respectively. Based on these two parameters, an additional useful parameter, the average point cloud density, can be calculated. The average point cloud density can be used as the resolution of the point cloud. This varies with the distance to the recording device, and this shows that the parameters of the quality model are chosen to be fundamental, so that optional parameter extensions are possible.

The segmentation tools require certain point cloud features to enable processing. Spectral features are often used to perform an automatic segmentation or to color the point for better visual differentiation. Most point clouds have geometric features (coordinates) and spectral features for color and intensity. In addition to these features, normals (N) are calculated to create perspective images or orient the single point within their neighborhood, as done with the *PCCT*. These features can enrich the point cloud after the semantic

segmentation. The exported feature can change during the semantic segmentation. The point clouds in the example are only extended by the feature semantic class. This is expressed by exporting each class as a single *pts* file. Closely related to the feature parameters of the point cloud is the file format that is available for import and export for software. The *pts* format is supported by all tools being used. This file format corresponds to the data model of Section 3.3. The used data model states that all segments should be available as an individual file. If the data model requires that the semantics of the point cloud have to be included in one file, then a different export file format must be used. This file format must have one additional space for the semantic label. The point clouds *Lab* and *Room* are currently not licensed and are only used internally, so there is no restriction on usage (P1.7). This means that the use of the datasets cannot be traced.

Table 12. Calculated and determined values for the quality parameters of availability and reliability of process. Object parameters with * are calculated in the segmentation software.

P. No.	Parameter Name	Lab RGB	Lab I	Lab R	Room RGB	Room I	Room R
P1	Availability						
P1.1	CD exists	yes	yes	yes	yes	yes	yes
P1.2	NoP		2,790,352 points			14,526,242 points	
P1.3	Area size		51 m ²			61 m ²	
P1.4	Object char. in.		x, y, z, I, R, G, B, xN*, yN*, zN*				
P1.5	Object char. out.		x, y, z, I, R, G, B, Class				
P1.6	File format out.	pts/csv	pts/csv	pts	pts/csv	pts /csv	pts
P1.7	Use restriction	no	no	no	no	no	no
P2	Reliability of Process						
P2.1	NoS	7	7	9	8	8	8
P2.2	ATR	13%	13%	55%	38%	45%	49%

In addition to point cloud metadata, metadata about the process are also represented by the process reliability, as shown in Table 12. Reliability can be determined if a process is performed independently multiple times. It can be determined by observing which parameters change systematically and which are random. According to the research question, two influences should be analyzed. On the one hand, the influence of different users is considered, and on the other hand, that of different tools is assessed. The repeat accuracy of different users is investigated in Section 4.1.4. At this point, the focus is on the two different tools. For a statistical consideration, the number of seven to nine annotations per tool is too small. However, a qualitative or comparative description of the influences of the tools in the form of a tendency is possible despite the small number of samples. For this purpose, the following values are not based on the annotations of individual annotators, but on a joint point cloud with all annotations. For the determination of the parameters of the datasets *Lab RGB* and *Lab I*, seven different annotations were performed; for the *Lab R* dataset, nine annotations were performed, and for the datasets *Room RGB*, *Room I* and *Room R*, eight annotations were performed.

The ATR is calculated based on the longest time for semantic segmentation for each point cloud. The maximum time is 120 min for the point cloud *Lab* and 194 min for the point cloud *Room*. For both point clouds, the semantic segmentation with *Recap* takes the longest. The ATR values in Table 12 show that the *PCCT* provides an average of only 13% of the maximum time for small point clouds such as *Lab*. With *Recap*, the ATR is 55% for the *Lab* point cloud. For the larger dataset, it can be seen that the *PCCT* can be used to work faster on average, but the differences in time decrease with increasing point cloud size.

The parameters of availability and process reliability are the basis on which to describe further parameters that have a more practical meaning for the investigated question. Thus far, it is described how a process can be carried out with the selected data and resources, how reliable this process and the other quality parameters are, as well as how efficient the tools and its usage are in comparison to others.

4.1.2. Integrity Characteristics

The integrity of the semantic point cloud is described by the parameters of the characteristics completeness, consistency and correctness, which are shown in Table 13. Completeness refers to the point cloud and its individual points. More precisely, it indicates how many points are still present after processing with a segmentation tool. After processing with *Recap*, the NoP was significantly reduced. The segmented point cloud still consists of 74% of the original points for the *Lab R* dataset and 41% for the dataset *Room R*. This point reduction is due to the tool. In other applications, this may arise from the task description—for example, if only $x\%$ of the point cloud is to be semantically segmented manually and the rest automatically. In addition to object completeness, semantic completeness can be determined. All classes described in the CD should exist in the semantic point cloud. This parameter is important for large and hierarchical CDs, when all levels are not or not yet classified. With respect to the segmentation tool, it must be possible to select or include the necessary classes. With *PCCT* and *Recap*, all five classes can be named and set with respect to the application. The point clouds are classified for all classes, but, for the following consideration, only the most detailed level is used. The furniture class is a super-class of the sub-classes table and chair. A super-class exists automatically if all sub-classes are present.

Section 4.1.1 describes which characteristics must be present for the point cloud. The presence of the characteristic is the necessary condition to evaluate whether the point clouds can be used. This is usually only possible if the point cloud features are consistent, which is the case for all six datasets, as shown in Table 13 by P4.1 to P4.3. The spectral features are scaled to the value range of 0 to 255 and the geometric features are given in meters.

A point cloud should have an equal amount of points for each class if it will be used as training data. The CE takes values of 0.65 (*Lab*) and 0.62 (*Room*), indicating that the class distribution is unequal (0.0 means equally distributed). The point cloud *Lab* consists of 90% of the floor class. The remaining 10% of points comprise chairs (3%), tables (6%) and scanning artifacts (1%). The distribution of the point cloud *Room* is comparable (Table 13).

Table 13. Calculated and determined values for the quality parameter of integrity.

P. No.	Parameter Name	Lab RGB	Lab I	Lab R	Room RGB	Room I	Room R
P3	Completeness						
P3.1	SSR	1.000	1.000	0.739	1.000	1.000	0.409
P3.2	NoC	5	5	5	5	5	5
P4	Consistency						
P4.1	GC x, y, z		10.34, 8.56, 1.00 m		8.07, 6.11, 0.82 m		
P4.2	SCRGB		0–255		0–255		
P4.3	SCI		0–255		0–255		
P4.4	CE	0.65	0.65	0.65	0.62	0.62	0.62
P5	Correctness						
P5.1	RP <i>floor</i>	99.9%	99.9%	100.0%	99.8%	99.9%	100.0%
P5.1	RP <i>chair</i>	96.1%	95.7%	99.2%	81.6%	66.0%	99.7%
P5.1	RP <i>table</i>	89.6%	89.6%	99.8%	94.5%	87.8%	99.7%
P5.1	RP <i>scan. artif.</i>	27.1%	27.6%	69.6%	47.1%	35.6%	77.2%
P5.2	RA <i>floor</i>	100.0%	100.0%	100.0%	99.8%	99.8%	100.0%
P5.2	RA <i>chair</i>	97.7%	97.1%	99.5%	97.7%	90.4%	99.7%
P5.2	RA <i>table</i>	96.8%	96.1%	99.8%	96.7%	95.9%	99.6%

The characteristic correctness can be determined if it is possible to describe what is true. This description can be made for a semantic point cloud by a semantically enriched geometry. The geometry either describes the target state from planning data or is captured and processed by a higher degree of correctness. This is the case if the point cloud was captured with a more accurate measurement system and a more accurate semantic segmentation method. For most furnished indoor scenes, no highly accurate geometric

planning data are available. In this work, a measurement and segmentation method is used that is significantly more accurate than the method under investigation. The method used for the creation of the semantic GT point cloud is based on simultaneous acquisition and semantic segmentation with the line scanning system *Leica T-Scan5*. The *Leica T-Scan5* is used in conjunction with the *Leica Lasertracker AT 960*. Based on the technical manufacturer specifications [97], the geometric accuracy (GA_P) of predominantly flat surfaces can be determined according to Equation (8).

$$GA_P = 80 \mu\text{m} + 3 \mu\text{m} * d \text{ m} (SD \text{ of } 2\sigma) \quad (8)$$

A maximum distance (d) between the laser tracker and the *Leica T-Scan5* of 10 m can be assumed. The maximum GA_P is therefore 0.11 mm. This can be set equal to the geometric correctness for the following consideration. The semantic is obtained by scanning the real objects individually with the *Leica T-Scan5* and assigning a semantic class during the measurement. Errors can occur due to the assignment of an incorrect class. This was minimized by intensive checks in the field and during data preparation (four-eyes principle). The original point clouds acquired with the *Leica T-Scan5* are further compressed and harmonized so that the maximum point density is less than 1 point/mm². The GT point clouds are considered free of semantic errors and contain only semantic objects. Scanning artifacts are not included in the GT point cloud. The point cloud in Figure 12 is the reference model for determining the correctness and the precision of the point cloud to be analyzed.

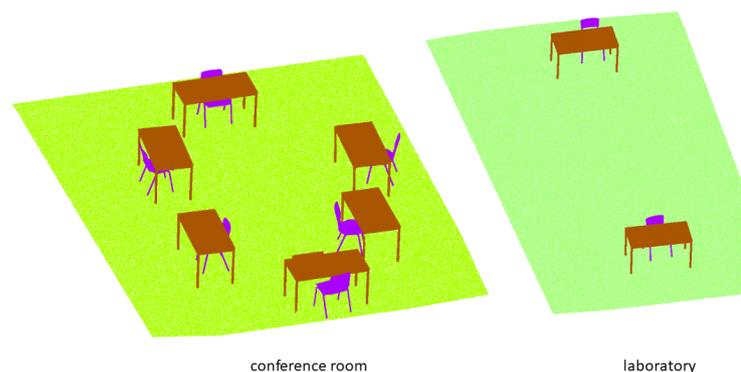


Figure 12. GT point cloud for determination and verification of correctness and precision parameters.

The determination of the correctness parameters can be performed if the GT and the analyzed point clouds are in the same coordinate system. Both point clouds are transformed via discrete target points into a local room coordinate system. Residuals of up to 7 mm (*Lab*) and 5 mm (*Room*) occur as a result of this transformation of the analyzed point cloud. The residuals are considered to denote uncertainty when comparing the point clouds to determine the quality parameter for correctness and precision.

The class segments of the GT point clouds are geometrically compared with those to be analyzed. For this comparison, the following rules apply:

- If the point distance between both point clouds is less than a threshold, then a point in the point cloud under investigation has been correctly semantically segmented. These points are TP points.
- If a segmented point in the investigated point cloud is closer to a segment of another class, then it is an FP point of the selected class.
- The FP points are also FN points of the other classes. By comparing the GT point cloud segments of the other classes with the sub-point cloud of the investigated point cloud, the FN points can be determined.

The resulting confusion matrix of TP, FP and FN points (Figure 7) provides the basis for determining the parameters RP and RA. These parameters express how correct a semantic segmentation is—RP by the ratio of the TP points to all points of the semantic target class

and RA by the ratio of the TP area to the total area of a semantic target class. The areas are calculated via a triangular meshing with the *Ball Pivoting Algorithm* by [98]. Depending on the application, either the RP or RA is more appropriate. RP is more meaningful for applications in which the individual points are important. This is the case if ML applications have to be validated. In these applications, it should be checked how well a task is solved with a dataset. For the use of a point cloud as a model or as a basis for modeling with parametrized geometries, the RA is more suitable.

All correctness parameters in Table 13 refer to a joint point cloud, which was calculated from all segmented point clouds of each dataset. A small program based on *Open3D* functions [99] was used for this purpose. The class membership of each point in the joint point cloud is based on the majority of the classifications within a dataset. The performance of individual annotations can be found in Tables A6–A8 in Appendix C.

The RP varies between 27.1% and 100.0%. The floor class is best recognized, with 99.9% to 100.0%. The chair and table classes were determined differently depending on the tools. For the *Lab R* and *Room R* datasets, the RP is higher than 99.1%. The RP of the *PCCT* datasets varies for the smaller semantic objects between 66.0% and 96.1%. It can be seen that semantic segmentation is better for the smaller point cloud *Lab* (PR higher than 89.6%) than for the larger *Room* point cloud (RP higher than 66.0%). The scanning artifacts are predominantly not detected in the segmentation with the *PCCT* (PR less than 47.1%). Moreover, with *Recap*, these classes are determined poorly, with a PR of only 69.6% and 77.2%, respectively (Table 13).

The RA is determined only for the object classes, since scanning artifacts are not useful for the visualization of an area. For all datasets, this parameter is higher than 90.3%. For the floor class, it is even higher than 99.7%. Since this parameter is based on the same data as the RP, similar behavior can be expected. However, differences occur due to the different point densities. For example, the RA is higher than the RP for all *PCCT* datasets, since this tool is used for areal segmentation and small groups of points (e.g., at class boundaries) are more often assigned to an incorrect class. This can be observed, e.g., in dataset *Room I*, with an RP for the chair class of 66.0% and with 87.8% for the table class. Here, the RA is 90.4% for the chair class and 95.9% for the table class (Table 13). The differences between RP and RA are smaller or do not occur for *Recap*, because the segments can be formed more finely and individually.

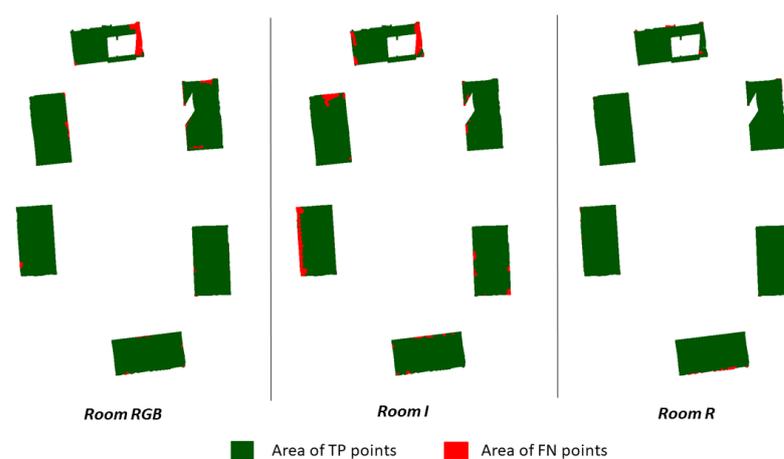


Figure 13. TP and FN areas of dataset *Room* at different semantic segmentations for the table class.

The analysis of the areas can also be useful, if the inverse RA, the area of FN points, is considered. This indicates which areas are not assigned to the correct class. These are holes or missing parts in the segmented point cloud. By visualizing these areas, it is possible to identify certain problematic sections for which the applied tools do not allow correct class assignment. The problematic sections are colored red in Figure 13.

4.1.3. Accuracy Characteristics

The accuracy in the quality model is expressed by the quality characteristics of precision and semantic accuracy. Precision is described by two ratio parameters. Additionally, MD and SD are determined from the FP points. The geometric accuracy is described for the handling of the semantic definitions and in terms of implementation per class (Table 14).

The PP of the floor class is higher than 99.5% for all datasets, so that all segmentation methods work equally well for this class. Based on the PA, the maximum incorrect area can also be determined with 0.5% of the object areas. The use of points or areas leads to no measurable differences.

In contrast, the semantic augmentations applied for the chair and table classes show varying precision. The semantic segments with *Recap* for chair and table consist of more than 95.4% of TP points and 95.6% of the TP area. Thus, only 0.23 m² of the table area and 0.17 m² of the chair area is falsely semantically segmented. The proportion of object class points in the scanning artifacts class is very small, with 4.6% and 2%. In points, this corresponds to approximately 125,000 and 290,000, respectively. The MD from the GT geometry is up to 92 mm. The SD of FP points is less than 39 mm.

The floor was determined by a plan fit; tables and chairs were segmented free-hand. It can be observed that more precise work can be achieved via free-hand segmentation. For the table and chair classes, the SD of FP points varies between 9 mm and 16 mm (Table 14). The PCCT segmentation of the two point clouds is less precise. The PP for chairs varies between 67.3% and 78.5%. In terms of surfaces, the PA varies between 90.4% and 91.7%. For the table class, the PP varies between 92.1% and 93.6%. The table class has a wider range for PA, from 87.7% to 97.6%. The PA of the chair class is considerably higher than the PP. For the table class, a higher PP can be determined for the dataset *Lab*. For dataset *Room*, the PA is lower and can be expressed as the area. For the table class, up to 1118 mm², and for the chair class, up to 380 mm² are incorrectly segmented. The proportion of object points in the scanning artifacts class is very high, which is expressed by the PP, which ranges from 30.8% to 61.3% for the PCCT datasets. This agrees with the observations on the RP for the object classes in Section 4.1.2.

An influence due to the segmentation with *RGB* or *I* values cannot be observed. However, it can be seen that the smallest object class, chair, is less precise for the dataset *PCCT I*. A comparison of the values in Table 14 shows that the PP and the PA are influenced by the size and content of a point cloud. As an example, this can be observed by the table class for the datasets *Lab RGB* and *Room RGB*. For *Lab RGB*, the PA is higher than the PP. For the dataset *Room RGB*, the PA is 5.5% lower than the PP. This occurs for the PCCT datasets, because scanning artifacts are present behind the objects. The scanning artifacts are often assigned to object classes at the front. This can be seen in Figure 14.

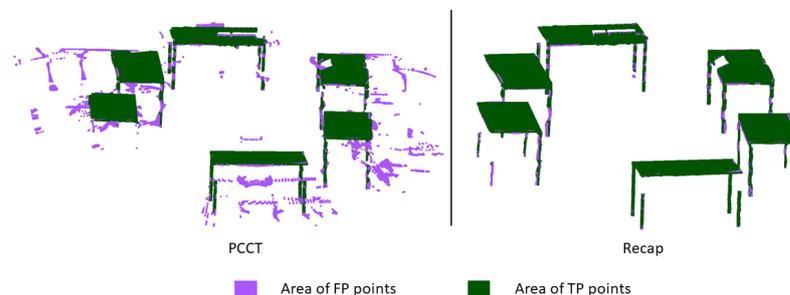


Figure 14. TP and FP points for the table class. The point cloud was semantically segmented with PCCT (left) and Recap (right).

The more scanning artifacts are spread, the more the PA of the object is affected. This can be observed with the parameter MD of FP points. The MD of FP points values are shorter for the *Lab RGB* than for the *Room RGB* dataset. This is also true for the SD of FP points, which is larger than 600 mm for the *Room RGB* dataset. The geometry of the class segments

becomes describable via the *SD and MD of FP points*. Based on the parameters PP and PA, it could be assumed that all point clouds are well-suited as a model. However, this is not the case due to the high deviations caused by scanning artifacts and overlapping segments in the PCCT datasets. The *SD of FP points* can describe, without visualization or human interpretation of the point cloud, that a point cloud is suitable or not as a model. A point cloud can be advantageous as a basis for modeling if the *MD of FP points* is large and the *SD of FP points* is small. These observations indicate isolated outliers. With SD and MD of FP points in combination, the quality of semantic point cloud segments can also be described geometrically.

The semantic accuracy can only be determined if the dataset-specific CD was used, since there is no general one. For these descriptions, the CD in Appendix A is applied. For all *Lab* and *Room* datasets, the CD was applied during all annotations. For other datasets, the respective CD of the dataset must be used. For our examples, the used CD is hierarchical, as can be seen in Table 14. In the CD, there are two semantic levels, which are filled out. Moreover, the class segments according to the CD are present or can be created by merging sub-classes into one super-class. Proof of the correct semantic class can be obtained by comparison with a reference or a visual inspection, as in Figure 8. For the example datasets, the furniture class cannot be seen directly, but it can be formed from the table and chair classes. One analytical strategy may be looking only at the most detailed classes. Since the furniture class is not directly present in our datasets, this class is not examinable and the parameter P7.3 is set to *no* in Table 14.

Table 14. Calculated and determined values for precision and semantic accuracy.

P. No.	Parameter Name	Lab RGB	Lab I	Lab R	Room RGB	Room I	Room R
P6	Precision						
P6.1	PP <i>floor</i>	99.7%	99.8%	99.8%	99.6%	99.6%	99.9%
P6.1	PP <i>chair</i>	78.5%	78.2%	95.8%	77.9%	67.3%	95.5%
P6.1	PP <i>table</i>	92.1%	93.1%	98.2%	93.2%	93.6%	97.5%
P6.1	PP <i>scan. artif.</i>	53.4%	52.6%	95.4%	61.3%	30.8%	98.0%
P6.2	PA <i>floor</i>	99.8%	100.0%	100.0%	99.5%	99.5%	99.5%
P6.2	PA <i>chair</i>	91.7%	91.6%	96.7%	90.4%	90.8%	95.7%
P6.2	PA <i>table</i>	96.8%	97.6%	98.8%	87.7%	87.6%	97.3%
		mm	mm	mm	mm	mm	mm
P6.3	MD FP pts <i>floor</i>	249	666	83	131	884	92
P6.3	MD FP pts <i>chair</i>	1278	1244	48	1699	1485	55
P6.3	MD FP pts <i>table</i>	1237	1558	53	1906	1967	53
		mm	mm	mm	mm	mm	mm
P6.4	SD FP pts. <i>floor</i>	42	87	38	61	161	24
P6.4	SD FP pts. <i>chair</i>	151	152	9	646	591	16
P6.4	SD FP pts. <i>table</i>	207	214	14	279	443	14
P7	Semantic Accuracy						
P7.1	CD applied	yes	yes	yes	yes	yes	yes
P7.2	Hier. CD	yes	yes	yes	yes	yes	yes
P7.3	<i>floor</i> used	yes	yes	yes	yes	yes	yes
P7.3	<i>furniture</i> used	no	no	no	no	no	no
P7.3	<i>chair</i> used	yes	yes	yes	yes	yes	yes
P7.3	<i>table</i> used	yes	yes	yes	yes	yes	yes
P7.3	<i>scan. artif.</i> used	yes	yes	yes	yes	yes	yes

4.1.4. Descriptive Use for Multiple Annotations

Regarding the research question, the individual annotation performance is also of interest. To investigate this aspect, 47 independent segmentations from nine different annotators are used for two point clouds. The metadata of the point cloud do not change due to the individual annotations, so only eight parameters describe the annotation differences. These are ATR, RP, RA, PP, PA, *MD of FP points*, *SD of FP points* and '*class*' used.

The processing time is determined in relation to the maximum time required and is presented for each annotation in Table A2 in Appendix B. An analysis of the individual segmentations shows that the largest differences in processing time occur for *Recap*. The fastest annotation has been performed with only 20% of the maximum duration for *Lab* resp. with 15% for *Room* by *PCCT*. For the semantic segmentations with the *PCCT*, the segmentation duration varies by 5% for *Lab I*, by 10% for *Lab RGB*, by 31% for *Room I* and by 39% for *Room RGB*. It can be seen from the values in Table A2 that the processing time is more consistent with *PCCT* than with *Recap* for different annotators. The longest semantic segmentation with *PCCT* was 38% faster than with *Recap*. Thus, *PCCT* has the advantage of a shorter processing time and better planning capability for tasks.

Further differences for the individual annotations can be found for the characteristics of correctness and precision. The parameters *RP* and *PP* are calculated for each segmentation (Tables A3–A8 in the Appendix C). Based on the small variation in all values for *RP* and *PP* for the floor class, it can be concluded that this class can be segmented very reliably, correctly and precisely using a geometry fit.

For the table and chair classes, the individual results are different. The correctness and the precision vary strongly. For *Recap*, the minimum *RP* is 49.6% and the maximum is 99.8%. The lowest *PP* value is 87.5% and varies up to 14%. It can be seen that the reliability for chairs and tables decreases, because different annotations reach different accuracies. The class with the lowest correctness is the scanning artifacts class (*RP* of max. 80.1%). The worst annotation for scanning artifacts with *Recap* contains only 42.3% *TP* points. Similar trends can be seen for the datasets processed with *PCCT*, but these are even lower in terms of precision and correctness.

To investigate whether the different results in a multiple segmentation occur by random or whether there is a systematic effect, we tested whether the set of the *RP* and the *PP* per class is *normally* or *t* distributed. The hypothesis is that the *RP* or the *PP* is *normally* distributed around an expected (average) value per class; thus, the annotation performance would then also be *normally* distributed. Random differences would be describable in this way. The *Kolmogorov–Smirnov test* [100] was used to test this hypothesis.

It was found that, for most classes of the *Room* datasets, the *RP* and *PP* are *normally* distributed. For the smaller *Lab* dataset, no *normal* distribution could be observed. The hypothesis can therefore not be confirmed. A possible reason for the different distributions could be that the larger point cloud has more random segmentation errors than the smaller point cloud, which is reflected in the parameters. In the small point cloud, the operator is more focused and the assignments are less ambiguous. This observation is supported by the fact that the *RP* and *PP* of the *Lab* datasets are predominantly higher.

The parameters *MD of FP points*, *SD of FP points*, *PA* and *RA* behave in a similar way to *RP* and *PP*, so these will not be discussed further. The parameter *class used* must be tested before joining to avoid gross errors in the joined point cloud. This can be tested during the joining by allowing only certain classes and excluding segmented point clouds that contain other classes.

4.1.5. Summary of the Descriptive Use

The description from Sections 4.1.1–4.1.3 focuses on comparing the tools and how they perform differently for smaller and larger point clouds. The basis of the investigation for each tool was a joined point cloud, which is free of individual segmentation patterns. It can be concluded that, with the quality model, semantic point clouds can be described for a comparison. Without further knowledge about the point cloud or the segmentation tool, an analysis of the point cloud can be performed based on 23 parameters. The quality model is holistic and does not only refer to parameters for correctness and precision, such as in [13,57]. *Recap* is more suitable than the *PCCT* for the outlined applications. Nonetheless, with the appropriate settings for the automatic segmentation, the *PCCT* is more efficient. The separation of objects and scanning artifacts has proven to be the main

problem. Based on the analysis process, and in connection with the developed tools, it is possible to investigate other segmentation tools.

The second part of the research question was discussed in Section 4.1.4. It can be noted that the processing time and the achieved accuracy are user-specific. There is no common relationship between long processing time and higher accuracy. However, it can be observed that, with *Recap*, a longer processing time leads to more accurate results in most cases. With *PCCT*, the processing time is, on average, 42% shorter. The influence of the user is noticeably large when using *Recap*. This can be seen in Table A4. For the same point cloud and tool, differences of up to 18% (PP) for object classes occur. This observation confirms the hypothesis that multiple processing is necessary, in order to allow a realistic evaluation of the quality of the point cloud.

4.2. Quality Model to Evaluate Semantic Point Clouds

The description of the semantic points from the previous Section 4.1 is the basis for an evaluation of a semantic point cloud. Due to the large number of semantic point clouds available on the web, it is difficult to obtain an overview of which point cloud is suitable for which application. The quality model, with its parameters, provides a framework for the comparison and selection of datasets. The parameters can be used to evaluate the characteristics of the point cloud in terms of metadata, geometry and semantics. Thus, the point clouds that do not meet the important criteria of an application can be excluded. In the following, the quality parameters for almost all datasets from Section 2.3 were researched. The research results were summarized in an Excel database. A threshold set and query functions were added. For the used thresholds, it can be queried whether they are met, not met or unknown. For the example semantic point cloud as a model, the query result is shown in Figure A1 of Appendix D. The public datasets in the database are extended by the datasets of the point clouds *Lab* and *Room*. For our own datasets, it is ensured that all parameters are known.

The collection of datasets shows that most of the metadata for the point clouds are sufficiently documented or can be determined from the datasets. Thereby, implicit parameters are derived. For example, a class definition is present, even if this is not written down, and can only logically be derived by an application or from the point cloud itself. In addition, it is concluded that at least one semantic segmentation took place. Therefore, the parameter NoS is assumed to be 1, if nothing else is found. Parameters concerning the size of the dataset, the file format and the data model can usually be taken from publications, web documentation or directly from the dataset.

The correctness and precision parameters are unknown for all external datasets. This is a central weakness of existing practice in dealing with datasets provided as training data or for modeling. This work tackles the problem by providing the quality model. The model should attempt to encourage the evaluation of published datasets (at least in part) for geo-semantic accuracy. This kind of evaluation is standard for automatic semantic segmentations in almost all publications. Since most automatic ML methods learn from human-annotated datasets that are not evaluated, these methods “learn” possible errors in the data. Thus, learning is done with a GT dataset, which is not always a true representation of reality. It is only the reality as seen (most of the time) by one annotator. In the end, only a relative evaluation of ML procedures is possible with currently available datasets.

The use of the Excel database does not aim to determine exactly one dataset for which all parameters are fulfilled. It should rather be an aid with which a selection can be made. Not all parameters are always relevant for all applications and can therefore be disregarded. A possible use of the quality model is now presented for the example application from above.

4.2.1. Point Cloud as Model

The semantic point cloud as a model is usually useful for an application for which the capturing sensor properties are well known and the point cloud has to be semantically

segmented at least once. The data model and the abstract model have to be known. The parameters of the quality characteristics of availability, process reliability, completeness, consistency and semantic accuracy must be fulfilled. The flowing example is a visualization of the floor, table and chair classes in CC.

The parameters CE and ATR have no meaning in this example, since no comparison of procedures is queried with regard to duration or training. The quality characteristics of correctness and precision are determined by one or more semantic segmentations, which always contain uncertainties. Thus, these parameters should never be set as 100%. Holes in the point cloud lower the correctness. Depending on how these holes occur and what additional information is available, the correctness can play a minor role. For the visualization, it is important that as few FP points as possible are present in the classes. This means that the precision must be high. The parameters RA and PA are favored over RP and PP in this application, since non-uniform density can be expected. Based on these considerations, we chose to set the thresholds for parameter RA to 70% and for the parameter PA to 80%. The scanning artifacts class is not considered, because it contains no object information. In addition, if PA is satisfied, the thresholds for *SD of FP points* and *MD of FP points* must be set to low values. Here, we suggest 50 mm as the threshold for *SD of FP points* and 100 mm as the threshold for *SD of FP points*. These limits vary from application to application. For a visual analysis of an indoor scene, our suggestions are sufficient to recognize objects such as tables and chairs.

The semantic conditions are fulfilled for the datasets *SceneNN*, *S3DSP* and *ScanNet* and our own datasets, *Lab* and *Room*. The *ScanNet* dataset is not available as a point cloud and therefore does not correspond to the research question. *SceneNN* and *S3DSP* are available in the appropriate file format for CC and have the necessary features (x , y , z -coordinates and semantic label). For an exclusive visualization, no restrictions of use are present. Subject to the unknown parameters, the datasets can be used for the example task.

For our own datasets, the semantic and all relevant formal constraints are satisfied (Section 4.1). The RA parameter for correctness and the PA parameter are satisfied for all classes as well, but the datasets processed with the *PCCT* do not meet the thresholds for *SD of FP points* and *MD of FP points*. The *PCCT* dataset cannot be used for the visualization. The *Recap* point clouds for *Lab* and *Room* meet the specifications and can be used. This is shown in Figure 15 for the point cloud *Room R*.

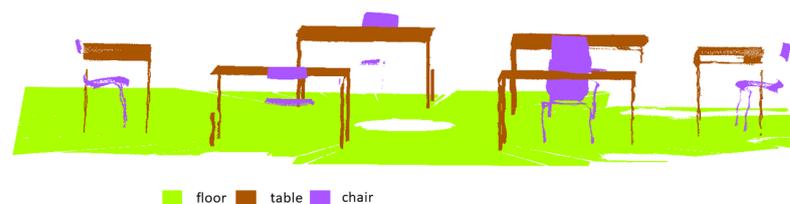


Figure 15. Semantic point cloud consisting of the floor, table and chair classes for visualization of a real room.

4.2.2. Point Cloud as a Basis for Modeling

A similar procedure as in Section 4.2.1 can be followed when using a semantic point cloud as the basis for modeling. The semantic parameters must also be fulfilled to model the needed classes. The semantic and geometric characteristics of the point clouds must be available. The file formats must be compatible with the modeling software. Most point clouds are available in open or open-source file formats that can be loaded with most modeling software, such as *PointCab* (<https://pointcab-software.com/en/> accessed on 15 December 2021). However, this is not always the case, as popular modeling programs, such as *Autodesk Revit* (<https://www.autodesk.de/products/revit/> accessed on 15 December 2021), only support their proprietary file formats. A software that supports open or open-source file formats should be used.

Due to the chosen semantic and formal target values, only the datasets *SceneNN* and *S3DSP*, as well as *Lab* and *Room*, can be considered for the example. Since no information is available for *SceneNN* and *S3DSP* regarding correctness and precision, these parameters cannot be evaluated. For our own datasets, *Lab* and *Room*, there are parameter values available, which are used to evaluate correctness and precision. As before, correctness is less important than precision as holes and incomplete edges can be closed or completed associatively when modeling with parametric geometry objects. In modeling by triangulation, holes can be closed up to a certain size. Thus, the threshold for correctness can be lowered to, e.g., RA 60% and complete modeling can still be achieved. The precision has higher relevance, because objects are mostly enlarged. The threshold for PA should remain at 80%. The thresholds for *SD* and *MD of the FP points* now have additional relevance as before. Distant single points are usually excluded automatically by the knowledge of the modeler, so this parameter can be very large (e.g., 2000 mm). More important is the *SD of the FP points*, which should remain at 50 mm. The choice of the threshold must also be customized for the task in question. For modeling objects using a model catalog or in LoD 100 or LoD 200 BIM applications, the proposed thresholds are sufficient. Due to the chosen threshold, only the two *Recap* datasets are available.

4.2.3. Point Cloud as Training Data

The third example is to use point clouds as information carriers to train data-based algorithms. For this purpose, the scanning artifacts class is necessary, in addition to the object classes from above. For many semantic indoor datasets, the scanning artifacts class or a comparable class for disturbances/noise is not included. For most outdoor datasets, not all indoor object classes are available.

Only our own datasets are considered in the following. The parameter NoP must be fulfilled, so that enough data for training and evaluation are available. A dataset with only 2 million points is too small for training. The training algorithm parameters will likely lead to unreliable and inaccurate results for other unknown point clouds. The target value for the NoP is set to 5 million points and at least three independent semantic segmentations are considered necessary to verify the knowledge in the data, even if no GT data of a higher accuracy level are available. The parameter ATR has the function of identifying, in the case of a large number of operations, the operators that work particularly fast. For example, these workers could be favored over the slower ones for further work.

The correctness and the precision for this work are equally important, because the method to be trained should learn the optimal handling of the data. Here, the points are the relevant input variables, which is why the RP and PP are used. The suggested thresholds are 75% for scanning artifacts and 80% for objects. It is expected that objects are segmented more distinctly and an interpretation of the scanning artifacts is more difficult. The other geometric parameters should not exceed the limits for the applications described above, but they are of minor importance for this application.

4.2.4. Summary of the Evaluated Use

The three example applications show how a semantic point cloud can be evaluated with the quality model and how it can be decided whether the quality of a dataset is sufficient. It should be emphasized that, with the parameters, an objective evaluation is possible, even if the relevance of the individual parameters is different in the respective application. The presented applications and used thresholds are only examples, based on our experience.

5. Conclusions and Outlook

Semantic 3D point clouds play a crucial role in the context of the digitization of working environments. A representation of reality as a detailed point cloud or in the form of a derived model is a fundamental component in many planning and management processes for buildings. Bringing semantic information into a geometric model is the

next major step towards the automation of planning and decision making. Integrating the semantics of objects as additional information into a point cloud is a necessary and challenging task that must be solved. The semantics of the point cloud must be describable in terms of resolution, correctness and precision. This requires additional metadata about the point cloud and the previous processes. The requirements of an application must be compared with the actual characteristics and it must be tested whether the requirements are fulfilled.

The quality characteristics of a point cloud can be described by a quality model. For the holistic description of a semantic point cloud, a model based on seven characteristics was deemed to be suitable, offering the user the possibility to describe, compare and evaluate their own as well as third-party point clouds. In order to describe the quality of semantic point clouds with a manageable number of parameters, a quality model was created and tested in this work. The choice of parameters was based on the underlying process, as well as on the abstract model and the data model.

The holistic quality model for semantic point clouds focused on the characteristics of semantic segmentation; the characteristics of geometric creation must also be taken into account. Crucial for the semantic segmentation are the accuracy and reliability with which a point cloud was split into semantic segments. In particular, the human influences on the GT point clouds are usually not considered. The initial semantic knowledge in a GT point cloud is always given by a human. The quality of the knowledge is a variable quantity. It depends on the motivation, training, perception and carefulness of the annotator. One way to keep these individual influences low is to use multiple independent annotators and a unique CD, and to train the annotators well. The use of different segmentation tools, as well as the degree of individualization, have a measurable impact on the final point cloud. The more individualization a tool allows, the better a single semantic segmentation can be. However, this has the disadvantage that the segmentation performance can vary.

The created quality model allows the comparison of publicly available semantic point cloud datasets. The analysis of a selection of publicly available point clouds has shown that, in particular, parameters for the GT correctness and GT precision are usually not provided and therefore a comparison is not possible. This is a central weakness, which has to be addressed in the current practice so that realistic semantics can be represented in a point cloud. Our quality model contributes to the improvement of GT point clouds.

In future, a distinction of the general model is necessary and an adaptation to data-based algorithms is to be recommended. The current quality model is only designed for indoor applications due to the complexity of the semantic environment and must be adapted for outdoor applications. It is conceivable that the data-based algorithms can be understood even better if the characteristics of the input data (point cloud) are described. Based on the determined characteristics of the input data and algorithm response, an objective performance comparison can be achieved.

Author Contributions: Conceptualization, E.B.; methodology, E.B.; software, E.B.; validation, E.B.; formal analysis, E.B.; investigation, E.B.; resources, E.B.; data curation, E.B.; writing—original draft preparation, E.B.; writing—review and editing, E.B. and H.S.; visualization, E.B.; supervision, H.S.; project administration, E.B.; funding acquisition, E.B. and H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The Excel database with 18 point clouds can be found at <https://github.com/eb17/Quality-check-of-point-cloud-data-sets> (accessed on 15 December 2021). Based on the database, the datasets can be selected according to usability. The developed quality model is implemented in this Excel table.

Acknowledgments: Many thanks to the annotators: Clemens Semmelroth, Stefanie Stand, Annette Scheider, Günter Eppinger, Sarah Lange, Friedrike Köpke, Mona Lütjens and Cigdem Askar. Special thanks to Stefanie for the support during the recording, to Clemens for the support during the evaluation and to Annette for proofreading.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Class Definition

The CD for the examples is shown in Table A1. A level is assigned to each class. A super-class is always fully divided into sub-classes. The class description is kept brief and provides relations between the classes.

Table A1. Class definition.

Level	Class	Definition
L0	furniture	<i>Furniture</i> includes objects that have contact with the floor and stand in the room. Objects that do not belong to the class <i>chair</i> or <i>table</i> cannot be <i>furniture</i> . The class can be further subdivided.
L1	table	The <i>table</i> class consists of all the points that describe/contain the table legs, the lower frame of the table, the <i>table</i> top and the adjustable feet.
L1	chair	The class <i>chair</i> consists of all the points that describe the seat, backrest, tubular frame and rubber feet.
L0	floor	The <i>floor</i> class consists of all points describing the flat floor and small edges and floor inlets (maintenance flaps). The <i>floor</i> can be considered a plane with a deviation of 50 mm.
L0	scanning artifacts	The <i>scanning artifacts</i> class consists of all points that describe objects lying on the ground—for example, cables. Furthermore, this includes all points that are caused by measurement errors (phantom points), reflection of the objects and gap closures due to the evaluation software. Multiple reflections can also occur.

Appendix B. Time Required for Semantic Segmentation

Table A2 shows the actual processing time in relation to the maximum processing time. The maximum time required in minutes for each point cloud is equal to 100%. The percentages can only be compared within one point cloud. The data are only valid for the comparison of the example described in Section 4.1.4. For other investigations, the ordinary times in minutes must be used.

Table A2. Actual processing time in relation to the maximum processing time.

No.	Lab			Room		
	RGB	I	R	RGB	I	R
1	12.5%	12.5%	41.7%	41.2%	46.4%	49.0%
2	14.2%	11.7%	64.2%	27.8%	33.5%	34.0%
3	12.5%	12.5%	100%	-	-	-
4	10.0%	10.8%	94.2%	32.5%	42.3%	100.0%
5	9.2%	-	35.8%	33.5%	51.5%	46.4%
6	19.2%	-	37.5%	23.2%	30.9%	30.9%
7	-	12.5%	75.0%	61.9%	61.9%	61.9%
8	-	15.8%	25.0%	46.1%	46.4%	15.5%
9	10.8%	13.3%	20.8%	41.2%	43.8%	51.5%
Avg.	13.0%	12.7%	54.8%	38.0%	44.6%	48.6%

Appendix C. Correctness and Precision for Multiple Annotations

Tables A3–A5 show the parameter PP for all classes of the CD from Appendix A. Tables A6–A8 contain the parameter RP of all annotations.

Table A3. Precision for the datasets *Lab* and *Room*, semantically segmented by *Recap*.

No.	Lab				Room			
	Floor	Chair	Table	Scan. Artif.	Floor	Chair	Table	Scan. Artif.
1	99.8%	95.9%	98.7%	91.7%	99.9%	96.4%	97.6%	89.7%
2	99.6%	91.7%	96.0%	97.8%	77.0%	99.8%	94.0%	98.5%
3	100.0%	85.9%	97.3%	17.7%	-	-	-	-
4	99.7%	97.9%	99.5%	88.2%	99.9%	95.6%	97.4%	87.5%

Table A3. Cont.

No.	Lab				Room			
	Floor	Chair	Table	Scan. Artif.	Floor	Chair	Table	Scan. Artif.
5	100.0%	93.4%	99.4%	47.4%	99.8%	94.1%	96.9%	77.5%
6	99.9%	97.0%	85.7%	81.1%	99.8%	94.3%	97.6%	33.9%
7	100.0%	96.6%	99.0%	74.3%	99.2%	94.7%	96.2%	82.7%
8	99.8%	99.2%	99.9%	76.4%	99.9%	92.7%	96.7%	90.4%
9	99.5%	99.8%	99.9%	77.6%	99.5%	90.9%	97.0%	95.0%

Table A4. Precision for the datasets *Lab* and *Room*, semantically segmented by *PCCT RGB*.

No.	Lab				Room			
	Floor	Chair	Table	Scan. Artif.	Floor	Chair	Table	Scan. Artif.
1	99.7%	93.9%	86.2%	40.5%	99.6%	77.9%	93.7%	51.3%
2	99.7%	88.5%	86.9%	41.7%	99.6%	79.1%	92.8%	43.7%
3	99.7%	83.3%	77.1%	29.9%	-	-	-	-
4	99.7%	94.6%	86.6%	41.0%	99.7%	80.9%	93.4%	61.4%
5	99.7%	91.6%	83.2%	47.4%	99.7%	82.7%	95.2%	41.7%
6	99.7%	84.2%	95.2%	24.0%	99.7%	77.3%	88.9%	43.4%
7	-	-	-	-	99.3%	74.0%	93.5%	48.5%
8	-	-	-	-	99.7%	78.4%	93.9%	56.0%
9	99.7%	93.9%	80.8%	38.1%	99.5%	80.4%	93.6%	49.9%

Table A5. Precision for the datasets *Lab* and *Room*, semantically segmented by *PCCT I*.

No.	Lab				Room			
	Floor	Chair	Table	Scan. Artif.	Floor	Chair	Table	Scan. Artif.
1	99.7%	93.8%	86.8%	38.5%	99.7%	73.9%	95.3%	26.7%
2	99.7%	93.8%	85.3%	39.8%	99.4%	82.1%	89.8%	42.0%
3	99.7%	81.9%	82.2%	21.7%	-	-	-	-
4	99.7%	93.6%	89.0%	37.8%	99.7%	77.7%	90.3%	33.3%
5	-	-	-	-	99.7%	82.7%	95.2%	41.7%
6	-	-	-	-	99.6%	76.8%	91.5%	30.1%
7	99.8%	77.0%	75.8%	37.7%	99.3%	70.7%	91.0%	20.6%
8	99.7%	94.2%	87.5%	38.0%	99.7%	77.3%	92.6%	35.5%
9	99.7%	83.5%	90.9%	35.5%	99.2%	75.7%	92.7%	30.8%

Table A6. Recall for the datasets *Lab* and *Room*, semantically segmented by *Recap*.

No.	Lab				Room			
	Floor	Chair	Table	Scan. Artif.	Floor	Chair	Table	Scan. Artif.
1	100.0%	98.9%	99.8%	62.0%	100.0%	98.0%	99.2%	75.9%
2	100.0%	99.9%	99.8%	46.4%	100.0%	97.6%	96.5%	76.2%
3	99.5%	90.7%	94.0%	80.1%	-	-	-	-
4	100.0%	98.3%	99.7%	67.0%	100.0%	99.0%	99.4%	74.0%
5	99.6%	98.5%	99.0%	69.8%	99.9%	99.1%	97.1%	66.7%
6	99.9%	70.4%	99.8%	74.8%	100.0%	49.6%	98.7%	71.1%
7	99.8%	97.8%	99.8%	79.1%	100.0%	97.9%	98.3%	42.3%
8	100.0%	96.4%	99.2%	77.1%	99.9%	99.3%	99.6%	64.4%
9	100.0%	96.0%	99.1%	57.2%	100.0%	99.4%	99.5%	51.1%

Table A7. Recall for the datasets *Lab* and *Room*, semantically segmented by *PCCT RGB*.

No.	Lab				Room			
	Floor	Chair	Table	Scan. Artif.	Floor	Chair	Table	Scan. Artif.
1	99.9%	80.5%	96.0%	26.3%	99.8%	83.0%	88.6%	54.4%
2	99.9%	84.6%	95.4%	19.3%	99.8%	68.2%	92.7%	46.6%
3	99.9%	83.6%	95.6%	20.4%	-	-	-	-
4	99.9%	84.8%	95.6%	22.2%	99.8%	80.4%	94.2%	55.9%
5	99.8%	84.6%	95.2%	18.1%	99.8%	50.8%	91.7%	64.0%
6	99.7%	93.5%	87.3%	32.6%	99.8%	69.7%	96.2%	29.2%
7	-	-	-	-	99.8%	68.2%	91.9%	44.2%
8	-	-	-	-	99.8%	87.1%	91.5%	52.7%
9	99.9%	84.8%	95.0%	15.2%	99.8%	77.2%	91.2%	49.4%

Table A8. Recall for the datasets *Lab* and *Room*, semantically segmented by *PCCTI*.

No.	<i>Lab</i>				<i>Room</i>			
	Floor	Chair	Table	Scan. Artif.	Floor	Chair	Table	Scan. Artif.
1	99.9%	83.8%	96.1%	22.0%	99.8%	46.2%	88.5%	54.7%
2	99.9%	83.2%	96.6%	20.0%	99.8%	57.1%	90.6%	45.3%
3	99.8%	85.1%	90.0%	29.6%	-	-	-	-
4	99.9%	84.3%	95.8%	23.6%	99.8%	80.4%	86.4%	39.0%
5	-	-	-	-	99.8%	74.7%	90.6%	52.0%
6	-	-	-	-	99.8%	65.4%	87.7%	38.0%
7	99.0%	84.0%	95.5%	15.1%	99.8%	56.4%	85.8%	26.5%
8	99.9%	83.7%	96.2%	21.6%	99.8%	81.4%	85.8%	43.5%
9	99.9%	87.6%	95.5%	24.3%	99.8%	61.4%	86.2%	37.6%

Appendix D. Point Cloud Dataset Comparison

Results

t = parameter is true
 nt = parameter is not true
 - = not enough information available

	Room RGB (PCCT)	Room I (PCCT)	Room R (Recap)	Lab RGB (PCCT)	Lab I (PCCT)	Lab R (Recap)	Paris-Lille-3D	Semantic3D	SemanticKITTI	MLS1 TUM City Campus	Toronto3D	CSPC-Database	BlipC-Database**	SceneNN	S3Dsp	ScanNet**	Matterport3D**	ScanObjectNv
P1.1 Class definition exists	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t
P1.2 Number of points	t	t	t	t	t	t	t	t	t	t	t	-	t	nt	-	-	-	t
P1.3 Area size	t	t	t	t	t	t	-	-	-	t	-	-	-	t	t	t	t	t
P1.4 Object characteristics in	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P1.5 Object characteristics out	t	t	t	t	t	t	t	t	t	-	t	t	-	t	t	t	t	t
P1.6 File format output	t	t	t	t	t	t	nt	t	nt	nt	nt	-	nt	nt	-	nt	t	-
P1.7 Use restriction	t	t	t	t	t	t	t	nt	nt	t	nt	nt	nt	nt	nt	nt	nt	nt
P2.1 Number of segmentation	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t
P2.2 Average time required	t	t	t	t	t	t	t	-	t	t	t	t	t	t	t	t	t	t
P3.1 Semantic segmentation rate	t	t	nt	t	t	nt	t	t	t	t	t	t	t	t	t	t	t	t
P3.2 Number of classes	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t
P4.1 Geometric consistency of geometry (x)	t	t	t	t	t	t	t	-	t	-	t	-	-	-	-	-	-	-
P4.1 Geometric consistency of geometry (y)	t	t	t	t	t	t	t	-	-	-	t	-	-	-	-	-	-	-
P4.1 Geometric consistency of geometry (z)	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P4.2 Spectral consistency of spectral RGB	t	t	t	t	t	t	t	t	t	t	t	-	-	-	-	-	-	-
P4.3 Spectral consistency of spectral I	t	t	t	t	t	t	t	t	t	t	t	-	-	-	-	-	-	-
P4.4 Class equality	t	t	t	t	t	t	-	-	-	-	t	t	-	-	-	-	-	-
P5.1 Recall points floor*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P5.1 Recall points chair*	t	nt	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P5.1 Recall points table*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P5.1 Recall points scan artifacts*	nt	nt	t	nt	nt	t	-	-	-	-	-	-	-	-	-	-	-	-
P5.2 Recall area floor*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P5.2 Recall area chair*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P5.2 Recall area table*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.1 Precision points floor*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.1 Precision points chair*	nt	nt	t	nt	nt	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.1 Precision points table*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.1 Precision points scan artifacts*	t	nt	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.2 Precision area floor*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.2 Precision area chair*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.2 Precision area table*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.3 Max. derivation FP points floor*	t	t	t	t	t	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.3 Max. derivation FP points chair*	nt	nt	t	nt	nt	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.3 Max. derivation FP points table*	nt	nt	t	nt	nt	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.4 Std. derivation FP points floor*	nt	nt	t	nt	nt	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.4 Std. derivation FP points chair*	nt	nt	t	nt	nt	t	-	-	-	-	-	-	-	-	-	-	-	-
P6.4 Std. derivation FP points table*	nt	nt	t	nt	nt	t	-	-	-	-	-	-	-	-	-	-	-	-
P7.1 Class definition applied	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t	t
P7.2 Hierarchical class definition	t	t	t	t	t	t	nt	t	nt	nt	t	t	t	nt	nt	nt	nt	nt
P7.3 Floor used*	t	t	t	t	t	t	nt	nt	nt	nt	nt	nt	nt	t	t	t	t	nt
P7.3 Chair used*	t	t	t	t	t	t	nt	nt	nt	nt	nt	nt	nt	t	t	t	t	t
P7.3 Table used*	t	t	t	t	t	t	nt	nt	nt	nt	nt	nt	nt	t	t	t	t	t
P7.3 Scan artifacts used*	t	t	t	t	t	t	nt	t	t	t	nt	t	nt	nt	nt	nt	nt	nt
P7.3 Furniture used*	nt	nt	nt	nt	nt	nt	nt	nt	nt	nt	nt	nt	nt	t	nt	nt	nt	nt

* Parameters to be adjusted depending on the number of classes.
 ** Data is mesh or image set

Figure A1. Point cloud dataset comparison. Example: Point cloud as model.

References

- Balangé, L.; Zhang, L.; Schwieger, V. First Step Towards the Technical Quality Concept for Integrative Computational Design and Construction. In *Springer Proceedings in Earth and Environmental Sciences*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 118–127. [\[CrossRef\]](#)
- Frangéz, V.; Salido-Monzú, D.; Wieser, A. Depth-Camera-Based In-line Evaluation of Surface Geometry and Material Classification For Robotic Spraying. In Proceedings of the 37th International Symposium on Automation and Robotics in Construction (ISARC), Kitakyushu, Japan, 27–28 October 2020; International Association for Automation and Robotics in Construction (IAARC): Berlin, Germany, 2020. [\[CrossRef\]](#)
- Placzek, G.; Brohmann, L.; Mawas, K.; Schwerdtner, P.; Hack, N.; Maboudi, M.; Gerke, M. A Lean-based Production Approach for Shotcrete 3D Printed Concrete Components. In Proceedings of the 38th International Symposium on Automation and Robotics in Construction (ISARC), Dubai, United Arab Emirates, 2–5 November 2021; International Association for Automation and Robotics in Construction (IAARC): Berlin, Germany, 2021. [\[CrossRef\]](#)
- Westphal, T.; Herrmann, E.M. (Eds.) *Building Information Modeling I Management Band 2*; Detail Business Information GmbH: München, Germany, 2018. [\[CrossRef\]](#)
- Hellweg, N.; Schuldt, C.; Shoushtari, H.; Sternberg, H. Potenziale für Anwendungsfälle des Facility Managements von Gebäuden durch die Nutzung von Bauwerksinformationsmodellen als Datengrundlage für Location-Based Services im 5G-Netz. In *21. Internationale Geodätische Woche Obergurgl 2021*; Wichmann Herbert: Berlin, Germany; Offenbach, Germany, 2021.
- Willemsen, T. Fusionsalgorithmus zur Autonomen Positionsschätzung im Gebäude, Basierend auf MEMS-Inertialsensoren im Smartphone. Ph.D. Thesis, HafenCity Universität Hamburg: Hamburg, Germany, 2016.
- Schuldt, C.; Shoushtari, H.; Hellweg, N.; Sternberg, H. L5IN: Overview of an Indoor Navigation Pilot Project. *Remote Sens.* **2021**, *13*, 624. [\[CrossRef\]](#)
- Grieves, M.; Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 85–113. [\[CrossRef\]](#)
- Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; IEEE: New York, NY, USA, 2015; pp. 922–928. [\[CrossRef\]](#)
- Hackel, T.; Wegner, J.D.W.; Schindler, K. Fast Semantic Segmentation of 3D Point Clouds with Strongly Varying Densit. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 177–184. [\[CrossRef\]](#)
- Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep Learning on Point Sets for 3d Classification and Segmentation. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: New York, NY, USA, 2017; pp. 77–85. [\[CrossRef\]](#)
- Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 2017; pp. 5099–5108. Available online: <https://arxiv.org/abs/1706.02413> (accessed on 15 December 2021).
- Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
- Zhu, J.; Gehrung, J.; Huang, R.; Borgmann, B.; Sun, Z.; Hoegner, L.; Hebel, M.; Xu, Y.; Stilla, U. TUM-MLS-2016: An Annotated Mobile LiDAR Dataset of the TUM City Campus for Semantic Point Cloud Interpretation in Urban Areas. *Remote Sens.* **2020**, *12*, 1875. [\[CrossRef\]](#)
- Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. Semantic3d.net: A New Large-scale Point Cloud Classification Benchmark. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *IV-1-W1*, 91–98. [\[CrossRef\]](#)
- Khoshelham, K.; Vilarinho, L.D.; Peter, M.; Kang, Z.; Acharya, D. The ISPRS Benchmark on Indoor Modelling. *ISPRS- Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *XLII-2/W7*, 367–372. [\[CrossRef\]](#)
- Padilla, R.; Passos, W.L.; Dias, T.L.B.; Netto, S.L.; da Silva, E.A.B. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit. *Electronics* **2021**, *10*, 279. [\[CrossRef\]](#)
- Rangesh, A.; Trivedi, M.M. No Blind Spots: Full-Surround Multi-Object Tracking for Autonomous Vehicles using Cameras and LiDARs. *IEEE Trans. Intell. Veh.* **2018**, *4*, 588–599. [\[CrossRef\]](#)
- Liu, X.; Qi, C.R.; Guibas, L.J. FlowNet3D: Learning Scene Flow in 3D Point Clouds. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: New York, NY, USA, 2019. [\[CrossRef\]](#)
- Wang, C.; Hou, S.; Wen, C.; Gong, Z.; Li, Q.; Sun, X.; Li, J. Semantic Line Framework-based Indoor Building Modeling Using Backpack Laser Scanning Point Cloud. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 150–166. [\[CrossRef\]](#)
- Volk, R.; Luu, T.H.; Mueller-Roemer, J.S.; Sevilimis, N.; Schultmann, F. Deconstruction Project Planning of Existing Buildings Based on Automated Acquisition and Reconstruction of Building Information. *Autom. Constr.* **2018**, *91*, 226–245. [\[CrossRef\]](#)
- Wang, C.; Dai, Y.; Elsheimy, N.; Wen, C.; Retscher, G.; Kang, Z.; Lingua, A. ISPRS Benchmark on Multisensory Indoor Mapping and Positioning. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *V-5-2020*, 117–123. [\[CrossRef\]](#)
- Bello, S.A.; Yu, S.; Wang, C. Review: Deep Learning on 3d Point Clouds. *Remote Sens.* **2020**, *12*, 1729. [\[CrossRef\]](#)

24. Liu, W.; Sun, J.; Li, W.; Hu, T.; Wang, P. Deep Learning on Point Clouds and Its Application: A Survey. *Sensors* **2019**, *19*, 4188. [CrossRef]
25. Xie, Y.; Tian, J.; Zhu, X.X. Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 38–59. [CrossRef]
26. Wang, X.; Zhou, B.; Shi, Y.; Chen, X.; Zhao, Q.; Xu, K. Shape2Motion: Joint Analysis of Motion Parts and Attributes from 3D Shapes. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: New York, NY, USA, 2019. [CrossRef]
27. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.
28. Omata, K.; Furuya, T.; Ohbuchi, R. Annotating 3D Models and their Parts via Deep Feature Embedding. In Proceedings of the 2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shanghai, China, 8–12 July 2019; IEEE: New York, NY, USA, 2019. [CrossRef]
29. Mo, K.; Guerrero, P.; Yi, L.; Su, H.; Wonka, P.; Mitra, N.; Guibas, L.J. StructureNet: Hierarchical Graph Networks for 3D Shape Generation. *ACM Trans. Graph.* **2019**, *38*, 1–19. [CrossRef]
30. Luhmann, T.; Robson, S.; Kyle, S.; Boehm, J. *Close-Range Photogrammetry and 3D Imaging*; De Gruyter: Berlin, Germany, 2013. [CrossRef]
31. Wasenmüller, O.; Stricker, D. Comparison of Kinect V1 and V2 Depth Images in Terms of Accuracy and Precision. In *Computer Vision—ACCV 2016 Workshops*; Springer International Publishing: Berlin/Heidelberg, Germany, 2017; pp. 34–45. [CrossRef]
32. Tölgyessy, M.; Dekan, M.; Chovanec, L.; Hubinský, P. Evaluation of the Azure Kinect and Its Comparison to Kinect V1 and Kinect V2. *Sensors* **2021**, *21*, 413. [CrossRef]
33. Schumann, O.; Hahn, M.; Dickmann, J.; Wohler, C. Semantic Segmentation on Radar Point Clouds. In Proceedings of the 2018 21st International Conference on Information Fusion, Cambridge, UK, 10–13 July 2018; IEEE: New York, NY, USA, 2018. [CrossRef]
34. Qian, K.; He, Z.; Zhang, X. 3D Point Cloud Generation with Millimeter-Wave Radar. *Proc. ACM Interactive Mob. Wearable Ubiquitous Technol.* **2020**, *4*, 1–23. [CrossRef]
35. Shults, R.; Levin, E.; Habibi, R.; Shenoy, S.; Honcheruk, O.; Hart, T.; An, Z. Capability of Matterport 3D Camera for Industria Archaeolog Sites Inventory. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W11*, 1059–1064. [CrossRef]
36. Sarbolandi, H.; Lefloch, D.; Kolb, A. Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect. *Comput. Vis. Image Underst.* **2015**, *139*, 1–20. [CrossRef]
37. Luhmann, T. *Nahbereichsphotogrammetrie Grundlagen-Methoden-Beispiele*; Wichmann: Berlin, Germany; Offenbach, Germany, 2018.
38. Freedman, B.; Shpunt, A.; Machline, M.; Arieli, Y. Depth Mapping Using Projected Patterns. U.S. Patent 2008/O2405O2A1, 3 October 2008.
39. Landau, M.J.; Choo, B.Y.; Beling, P.A. Simulating Kinect Infrared and Depth Images. *IEEE Trans. Cybern.* **2016**, *46*, 3018–3031. [CrossRef] [PubMed]
40. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3d Semantic Parsing of Large-scale Indoor Spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543. [CrossRef]
41. Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Nießner, M.; Savva, M.; Song, S.; Zeng, A.; Zhang, Y. Matterport3D: Learning from RGB-D Data in Indoor Environments. International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; IEEE: New York, NY, USA, 2017. [CrossRef]
42. Matterport. Matterport Pro 3D Camera Specifications. Available online: https://support.matterport.com/s/articledetail?language=en_US&ardId=kA05d000001DX3DCAW (accessed on 23 September 2021).
43. Hansard, M.; Lee, S.; Choi, O.; Horaud, R. *Time-of-Flight Cameras*; Springer: London, UK, 2013. [CrossRef]
44. Keller, F. Entwicklung eines Forschungsorientierten Multi-Sensor-System zum Kinematischen Laserscannings Innerhalb von Gebäuden. Ph.D. Thesis, HafenCity Universität Hamburg: Hamburg, Germany, 2015; ISBN 978-3844044171.
45. VelodyneLiDAR. Velodyne HDL-32E Data Sheet. Available online: https://www.mapix.com/wp-content/uploads/2018/07/97-0038_Rev-M_-HDL-32E_Datasheet_Web.pdf (accessed on 24 June 2021).
46. Riegl. RIEGL VZ-400-Data Sheet. Available online: www.riegl.com/uploads/tx_pxpriegldownloads/10_DataSheet_VZ-400_2017-06-14.pdf (accessed on 24 June 2021).
47. Lovas, T.; Hadzijanisz, K.; Papp, V.; Somogyi, A.J. Indoor Building Survey Assessment. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *XLIII-B1-2020*, 251–257. [CrossRef]
48. Kersten, T.P.; Lindstaedt, M.; Stange, M. Geometrische Genauigkeitsuntersuchungen aktueller terrestrischer Laserscanner im Labor und im Feld. *AVN* **2021**, *2*, 59–67.
49. ISO17123-9; Optics and Optical Instruments. Field Procedures for Testing Geodetic and Surveying Instruments. Terrestrial Laser Scanners. British Standards Institution: London, UK, 2018.
50. Kaartinen, H.; Hyypää, J.; Kukko, A.; Jaakkola, A.; Hyypää, H. Benchmarking the Performance of Mobile Laser Scanning Systems Using a Permanent Test Field. *Sensors* **2012**, *12*, 12814–12835. [CrossRef]
51. Wujanz, D.; Burger, M.; Tschirschwitz, F.; Nietzschmann, T.; Neitzel, F.; Kersten, T. Determination of Intensity-Based Stochastic Models for Terrestrial Laser Scanners Utilising 3D-Point Clouds. *Sensors* **2018**, *18*, 2187. [CrossRef] [PubMed]

52. Neuer, H. Qualitätsbetrachtungen zu TLS-Daten. *Qualitätssicherung geodätischer Mess-und Auswerteverfahren 2019. DVW-Arbeitskreis 3 Messmethoden und Systeme*; Wißner-Verlag: Augsburg, Germany, 2019; Volume 95, pp. 69–89.
53. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d ShapeNets: A Deep Representation for Volumetric Shapes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; IEEE: New York, NY, USA, 2015. [\[CrossRef\]](#)
54. Winiwarter, L.; Pena, A.M.E.; Weiser, H.; Anders, K.; Sánchez, J.M.; Searle, M.; Höfle, B. Virtual laser scanning with HELIOS++: A novel take on ray tracing-based simulation of topographic full-waveform 3D laser scanning. *Remote Sens. Environ.* **2022**, *269*, 112772. [\[CrossRef\]](#)
55. Iqbal, J.; Xu, R.; Sun, S.; Li, C. Simulation of an Autonomous Mobile Robot for LiDAR-Based In-Field Phenotyping and Navigation. *Robotics* **2020**, *9*, 46. [\[CrossRef\]](#)
56. Hua, B.S.; Pham, Q.H.; Nguyen, D.T.; Tran, M.K.; Yu, L.F.; Yeung, S.K. SceneNN: A Scene Meshes Dataset with aNnotations. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; IEEE: New York, NY, USA, 2016. [\[CrossRef\]](#)
57. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. Scannet: Richly-annotated 3d Reconstructions of Indoor Scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, US, 21–26 July 2017; IEEE: New York, NY, USA, 2017. [\[CrossRef\]](#)
58. Uy, M.A.; Pham, Q.H.; Hua, B.S.; Nguyen, D.T.; Yeung, S.K. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; IEEE: New York, NY, USA, 2019. [\[CrossRef\]](#)
59. CloudCompare. 3d Point Cloud and Mesh Processing Software Open-Source Project. Version 2.12. Available online: <http://www.cloudcompare.org/> (accessed on 24 June 2021).
60. Felzenszwalb, P.F.; Huttenlocher, D.P. Efficient Graph-based Image Segmentation. *Int. J. Comput. Vis.* **2004**, *59*, 167–181. [\[CrossRef\]](#)
61. Nguyen, D.T.; Hua, B.S.; Yu, L.F.; Yeung, S.K. A Robust 3D-2D Interactive Tool for Scene Segmentation and Annotation. *IEEE Trans. Vis. Comput. Graph.* **2018**, *24*, 3005–3018. [\[CrossRef\]](#) [\[PubMed\]](#)
62. Wada, K. labelme: Image Polygonal Annotation with Python. 2016. Available online: <https://github.com/wkentaro/labelme> (accessed on 15 December 2020).
63. Hossain, M.; Ma, T.; Watson, T.; Simmers, B.; Khan, J.; Jacobs, E.; Wang, L. Building Indoor Point Cloud Datasets with Object Annotation for Public Safety. In Proceedings of the 10th International Conference on Smart Cities and Green ICT Systems, Online, 28–30 April 2021; SciTePRESS—Science and Technology Publications: Setubal, Portugal, 2021. [\[CrossRef\]](#)
64. Roynard, X.; Deschaud, J.E.; Goulette, F. Paris-lille-3d: A Large and High-quality Ground-truth Urban Point Cloud Dataset for Automatic Segmentation and Classification. *Int. J. Robot. Res.* **2018**, *37*, 545–557. [\[CrossRef\]](#)
65. Tan, W.; Qin, N.; Ma, L.; Li, Y.; Du, J.; Cai, G.; Yang, K.; Li, J. Toronto-3D: A Large-scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020. [\[CrossRef\]](#)
66. Tong, G.; Li, Y.; Chen, D.; Sun, Q.; Cao, W.; Xiang, G. CSpC-Dataset: New LiDAR Point Cloud Dataset and Benchmark for Large-Scale Scene Semantic Segmentation. *IEEE Access* **2020**, *8*, 87695–87718. [\[CrossRef\]](#)
67. Zimmer, W.; Rangesh, A.; Trivedi, M. 3D BAT: A Semi-Automatic, Web-based 3D Annotation Toolbox for Full-Surround, Multi-Modal Data Streams. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; IEEE: New York, NY, USA, 2019. [\[CrossRef\]](#)
68. Ibrahim, M.; Akhtar, N.; Wise, M.; Mian, A. Annotation Tool and Urban Dataset for 3D Point Cloud Semantic Segmentation. *IEEE Access* **2021**, *9*, 35984–35996. [\[CrossRef\]](#)
69. Wirth, F.; Quehl, J.; Ota, J.; Stiller, C. PointAtMe: Efficient 3D Point Cloud Labeling in Virtual Reality. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; IEEE: New York, NY, USA, 2019. [\[CrossRef\]](#)
70. Monica, R.; Aleotti, J.; Zillich, M.; Vincze, M. Multi-label Point Cloud Annotation by Selection of Sparse Control Points. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; IEEE: New York, NY, USA, 2017. [\[CrossRef\]](#)
71. Autodesk-Recap. Youtube Channel. Available online: <http://https://www.youtube.com/user/autodeskrecap/> (accessed on 24 June 2021).
72. Barnefske, E.; Sternberg, H. PCCT: A Point Cloud Classification Tool To Create 3D Training Data To Adjust And Develop 3D ConvNet. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W16*, 35–40. [\[CrossRef\]](#)
73. *ISO9000*; Quality Management Systems—Fundamentals and Vocabulary. ISO: Geneva, Switzerland, 2015.
74. *DIN55350*; Concepts for Quality Management and Statistics—Quality Management. DIN: Geneva, Switzerland, 2020.
75. *DIN18710*; Engineering Survey. DIN: Geneva, Switzerland, 2010.
76. Blankenbach, J., Bauaufnahme, Gebäudeerfassung und BIM. In *Ingenieurgeodäsie: Handbuch der Geodäsie, Published by Willi Freuden and Reiner Rummel*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 23–53. [\[CrossRef\]](#)
77. Joos, G. Zur Qualität von Objektstrukturierten Geodaten. Ph.D. Thesis, Universität der Bundeswehr München, Muenchen, Germany, 2000.

78. Scharwächter, T.; Enzweiler, M.; Franke, U.; Roth, S. Efficient Multi-cue Scene Segmentation. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 435–445. [CrossRef]
79. Miller, G.A.; Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, K.J. Introduction to WordNet: An On-line Lexical Database. *Int. J. Lexicogr.* **1990**, *3*, 235–244. [CrossRef]
80. buildingSMART. Industry Foundation Classes 4.0.2.1. Available online: <https://standards.buildingsmart.org> (accessed on 24 June 2021).
81. BIM.Hamburg. *BIM-Leitfaden für die FHH Hamburg*; Technical Report; BIM: Hamburg, Germany, 2019.
82. Kaden, R.; Clemen, C.; Seuß, R.; Blankenbach, J.; Becker, R.; Eichhorn, A.; Donaubaue, A.; Gruber, U. Leitfaden Geodäsie und BIM. Techreport 2.1, DVW e.V. und Runder Tisch GIS e.V. 2020. Available online: <https://dvw.de/images/anhang/2757/leitfaden-geodaesie-und-bim2020onlineversion.pdf> (accessed on 15 December 2021).
83. BIM-Forum. Level of Development Specification Part1 & Commentary. 2020. Available online: <https://bimforum.org/lod/> (accessed on 15 December 2021).
84. Günther, M.; Wiemann, T.; Albrecht, S.; Hertzberg, J. Model-based furniture recognition for building semantic object maps. *Artif. Intell.* **2017**, *247*, 336–351. [CrossRef]
85. Wiltscho, T. Sichere Information Durch infrastrukturgestützte Fahrerassistenzsysteme zur Steigerung der Verkehrssicherheit an Straßenknotenpunkten. Ph.D. Thesis, University Stuttgart, Stuttgart, Germany, 2004.
86. Torralba, A.; Efros, A.A. Unbiased Look at Dataset Bias. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; IEEE: New York, NY, USA, 2011. [CrossRef]
87. Niemeier, W. *Ausgleichsrechnung*, 2nd ed.; De Gruyter: Berlin, Germany, 2008.
88. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2017.
89. Powers, D.M.W. Evaluation: From Precision, Recall and F-measure to Roc, Informedness, Markedness and Correlation. *Int. J. Mach. Learn. Technol.* **2017**, *2*, 37–63
90. Becker, R.; Lublasser, E.; Martens, J.; Wollenberg, R.; Zhang, H.; Brell-Cokcan, S.; Blankenbach, J. *Enabling BIM for Property Management of Existing Buildings Based on Automated As-is Capturing*; Leitfaden Geodäsie und BIM: Buehl, Germany; Muenchen, Germany, 2019. [CrossRef]
91. Engelmann, F.; Kontogiannia, T.; Hermans, A.; Leibe, B. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCV), Venice, Italy, 22–29 October 2017; IEEE: New York, NY, USA, 2017. [CrossRef]
92. Koguciuk, D.; Chechliński, Ł. 3D Object Recognition with Ensemble Learning - A Study of Point Cloud-Based Deep Learning Models. In *Advances in Visual Computing*; Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 100–114. [CrossRef]
93. Winiwarter, L.; Mandlbürger, G.; Pfeifer, N. *Klassifizierung von 3D ALS Punktwolken mit Neuronalen Netzen*; 20. Internationale Geodätische Woche Obergurgl 2019; Wichmann Herbert: Berlin, Germany; Offenbach, Germany, 2019; Volume 20.
94. Reiterer, A.; Wäschle, K.; Störk, D.; Leydecker, A.; Gitzen, N. Fully Automated Segmentation of 2D and 3D Mobile Mapping Data for Reliable Modeling of Surface Structures Using Deep Learning. *Remote Sens.* **2020**, *12*, 2530. [CrossRef]
95. Zoller+Fröhlich-GmbH. *Reaching New Levels, Z+F Imager5016, User Manual, V2.1*; Zoller & Fröhlich GmbH: Wangen im Allgäu, Germany, 2019.
96. Neitzel, F.; Gordon, B.; Wujanz, D. DVW-Merkblatt 7-2014, Verfahren zur Standardisierten Überprüfung von Terrestrischen Laserscannern (TLS). Technical Report, DVW. Available online: <https://dvw.de/veroeffentlichungen/standpunkte/1149-verfahren-zur-standardisierten-ueberpruefung-von-terrestrischen-laserscannern-tls> (accessed on 28 October 2021).
97. HexagonMetrology. Product Brochure Leica T-Scan TS 50-a. Available online: https://w3.leica-geosystems.com/downloads123/m1/metrology/t-scan/brochures/leica%20t-scan%20brochure_en.pdf (accessed on 24 June 2021).
98. Bernardini, F.; Mittleman, J.; Rushmeier, H.; Silva, C.; Taubin, G. The Ball-pivoting Algorithm for Surface Reconstruction. *IEEE Trans. Vis. Comput. Graph.* **1999**, *5*, 349–359. [CrossRef]
99. Zhou, Q.Y.; Park, J.; Koltun, V. Open3D: A Modern Library for 3D Data Processing. *arXiv* **2018**, arXiv:1801.09847v1.
100. Hodges, J.L. The Significance Probability of the Smirnov Two-sample Test. *Ark. Mat.* **1958**, *3*, 469–486. [CrossRef]

A.3 Evaluation of Class Distribution and Class Combinations on Semantic Segmentation of 3D Point Clouds with PointNet

Reference:

E. Barnefske & H. Sternberg (2023): Evaluation of Class Distribution and Class Combinations on Semantic Segmentation of 3D Point Clouds with PointNet, IEEE Access, 11, pp. 3826–3845. DOI: 10.1109/ACCESS.2022.3233411.

Graphical Abstract:

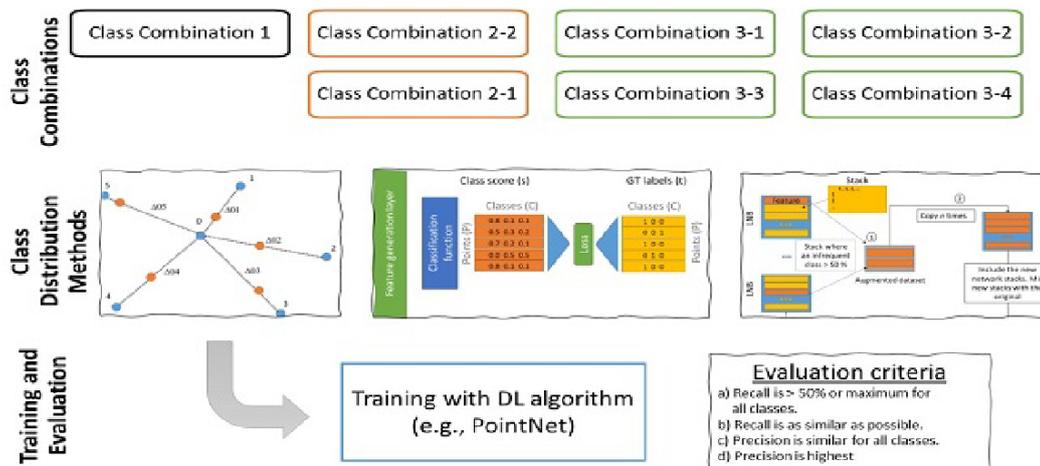


Figure 43: Graphical Abstract: The point clouds are separated into different semantic combinations for the training (first row). Different methods are used to extend the class distribution (second line). A DL algorithm is used to train the combinations and extensions, which are then evaluated according to fixed evaluation criteria.

Contribution of Co-Authors:

Table 8: Contribution to Paper No. 3

Involved in	Estimated contribution
Ideas and conceptual design	95%
Computation and results	100%
Analysis and interpretation	100%
Manuscript, figures and tables	100%
Total:	99%

I hereby confirm the correctness of the declaration of the contribution of Eike Barnefske for Paper 3 in Table 8:

Prof. Dr.-Ing. Harald Sternberg, HafenCity Universität Hamburg

Received 8 December 2022, accepted 25 December 2022, date of publication 30 December 2022, date of current version 25 December 2022

Digital Object Identifier 10.1109/ACCESS.2022.3233411

Evaluation of Class Distribution and Class Combinations on Semantic Segmentation of 3D Point Clouds with PointNet

EIKE BARNEFSKE¹ and HARALD STERNBERG¹

¹HafenCity University Hamburg, 22335 Hamburg, Germany

Corresponding author: Eike Barnefske (e-mail: eike.barnefske@hcu-hamburg.de).

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

ABSTRACT Point clouds are generated by light imaging, detection and ranging (LIDAR) scanners or depth imaging cameras, which capture the geometry from the scanned objects with high accuracy. Unfortunately, these systems are unable to identify the semantics of the objects. Semantic 3D point clouds are an important basis for modeling the real world in digital applications. Manual semantic segmentation is a labor and cost intensive task. Automation of semantic segmentation using machine learning and deep learning (DL) approaches is therefore an interesting subject of research. In particular, point-based network architectures, such as PointNet, lead to a beneficial semantic segmentation in individual applications. For the application of DL methods, a large number of hyperparameters (HPs) have to be determined and these HPs influence the training success. In our work, the investigated HPs are the class distribution and the class combination. By means of seven combinations of classes following a hierarchical scheme and four methods to adapt the class sizes, these HPs are investigated in a detailed and structured manner. The investigated settings show an increased semantic segmentation performance, by an increase of 31% in recall for the class Erroneous points or that all classes have a recall of higher than 50%. However, based on our results the correct setting of only these HPs does not lead to a simple, universal and practical semantic segmentation procedure.

INDEX TERMS 3D point clouds, Data hyperparameter, Hierarchical class combination, Hyperparameter, PointNet, Semantic classes, Semantic Segmentation, Unbalanced data

I. INTRODUCTION

SCENES of the real world are scanned with depth imaging cameras and light imaging, detection and ranging (LIDAR) scanners in a short time with high geometric resolution and accuracy [1]. The digitized scenes are mostly unsorted, unstructured and incomplete point clouds [2], [3], which form the basis of a geometric model. These kind of models are useful in a wide variety of applications such as, urban planning, tourism marketing, indoor navigation, robotic control, autonomous driving, building construction planning, building operation, heritage preservation, archaeological investigations, forestry and agriculture, or infrastructure maintenance [4]–[8]. The creation of these models is often done by hand, because humans are excellent

at interpreting visualized 3D point clouds and identifying semantic objects within them. Automated modeling by an algorithm requires that each point carries semantic features that can be used to form discrete semantic objects in a scene. Extending the point cloud with semantic features is semantic segmentation. The automated semantic segmentation is often performed by Machine Learning (ML) and Deep Learning (DL) approaches, which are a current research topics [4], [7], [9], [10].

DL-methods for semantic segmentation of 2D images achieve very high accuracies, but cannot be simply applied to point clouds due to the above mentioned properties. Many approaches exist where the point cloud is first transformed into an order and structure [11], [12]. However, point-based

methods such as PointNet [13] or RandLA-Net [14] omit this step and can perform a semantic segmentation directly from the original point cloud. In order to use these semantically segmented point clouds to create a building information model (BIM), the semantic point segments must meet certain accuracy requirements that arise from the model specifications [15]. These accuracy requirements are defined in the Level of Accuracy (LoA) [16], Level of Detail (LoD) [17] or the Level of Development (LoDev) [18]. For a BIM of the level *LoA2* (15 mm to 500 mm) or *LoDev 200* (design planning) and higher, the point cloud segments often cannot fulfill the geometric or semantic requirements, so that an improvement of the semantic segmentation step is necessary. Considering the complexity of the point cloud datasets, the automatic semantic segmentation is a key processing step for an efficient modeling.

Increased accuracy of these semantic segmentation methods is possible with training data [19] and Hyperparameters (HPs) [20], in addition to the adaptation of the network architecture. HPs are selected before training and commonly prior knowledge is used for the selection. They control and influence the training progress [20]. In this work the influence of the HPs *Unbalanced class distributions* and different *Class combinations* are investigated using the established network architecture PointNet (Section IV). For this purpose, four data- or algorithm-based methods for harmonizing class sizes are applied and adapted. In addition, a hierarchical class definition for frequent classes in a BIM is developed and applied. Further central contributions of this work are:

- A review of HP determination methods, data augmentation methods, and hierarchical semantic segmentation methods (Section II).
- The creation of a new medium-sized dataset that is suitable for BIM applications (Section III).
- The systematic evaluation of data augmentation methods and of hierarchical class combinations (Section V and VI).

All findings are summarized in Section VII and an outlook on further investigations is given.

II. STATE OF THE ART

A ML model is influenced by a large number of HPs. One key challenge when working with complex ML methods is to fully capture these HPs and to define the optimal values for them, as investigated by [21]–[24]. In Fig. 1, the relevant HPs for semantic segmentations are grouped into six clusters. The top row represents general HPs that relate to network architecture, regulation, optimization, and initialization [25]. In the bottom row (area with the blue background), Data Hyperparameters (DHPs) are shown. The DHPs depend on the data characteristics and not on the chosen model.

DHPs can be distinguished according to semantic, structural, geometrical and spectral characteristics of the dataset. The semantic characteristics of point clouds are described by [26]–[28]. In terms of structural characteristics, the definition of point neighborhood [29], data augmentation [30], and

the unbalanced class distribution for training data in general (e.g., images) [31] are topics that have already been investigated in other studies. Generally, geometrical and spectral features of point clouds are often used as training data by manual augmentation of point feature spaces [32]. These hand drawn features are point normals, eigenvalues, density values or mixed features [33]–[35]. So far, only few studies on the unbalanced class distribution and hierarchic semantic segmentation in point clouds are published. An overview of them is presented in Sections II-B and II-C.

A. POINTNET

DL-models for semantic segmentations of point clouds are usually distinguished by the input formats into which the point cloud is transformed. A categorization is presented in [36]. In their work, a categorization is made into discretization-based / structure-based (e.g., as voxel), projection-based (e.g., 2D-image), and point-based (e.g., raw points or graph) methods, which can further refined (Fig. 2). While initially discretization-based [37]–[39] and projection-based methods [40] were predominantly used, nowadays most of the (non-real-time) models are point-based [41]. Point-based methods use the unordered points themselves to perform semantic segmentation.

One of the most widely used point-based method is PointNet [13]. PointNet addresses the structural disadvantage of the point cloud format when processing them with DL-methods. This means that points do not have to be placed in a fixed order prior to processing. They can be arranged free in orientation and position in space.

The full functionality of PointNet is explained in the first published article from the developers [13] and in many reviews such as [43], [44]. In the following, the central processing steps of PointNet are presented for a better understanding of our investigations. Furthermore, the limitations of PointNet will be outlined.

1) Processing steps of PointNet

Processing with PointNet can be divided into three main steps. In the first processing step, the features are transformed into a uniform n -dimensional space using an affine transformation (with the T-Net module of PointNet). The transformation parameters are learned by the network. This transformation ensures that all input blocks are nearly at the same position and almost have the same orientation. An example with a point cloud of a chair is given in Fig. 3. This transformation is repeated after the first extraction of depth features, so that the depth features are also aligned in the complex feature space (e.g., 64 dimensions) [13].

The second processing step is the extraction of depth features-based on the input features (e.g., 3D coordinates, point normals or color values) or previous depth features. This is done using different transformation layers or a multi-layer-perceptron [13]. In most implementations of PointNet, a 1D or a 2D convolutional layer is used. As shown in Fig. 4a, the rows of the tensor are equal to the number of

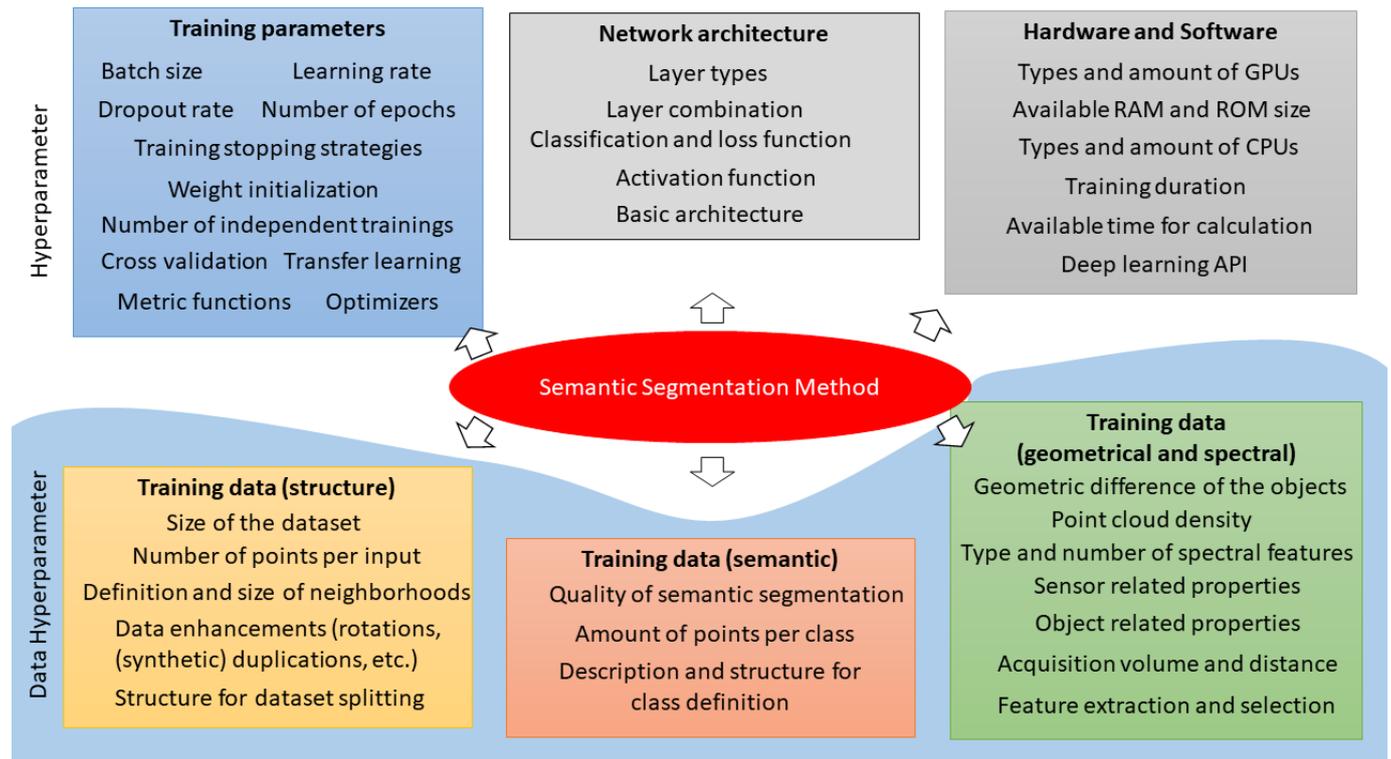


FIGURE 1. Influencing variables and parameters for the development of a DL-based semantic segmentation method. The parameters and influencing variables shown are a selection and might be adapted for other applications.

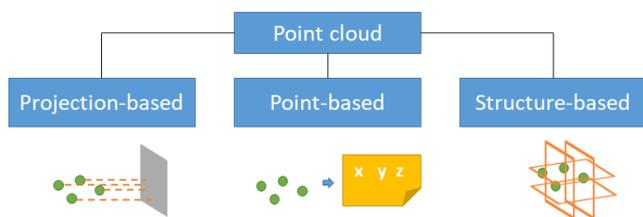


FIGURE 2. Preparation of point clouds for semantic segmentation with DL, by projection into image space, organization into a 3D structure, and usage of the raw point cloud. (Figure taken from [42] and adapted.)

block points and only one column is occupied. The features of the points are arranged in the depth layer of the tensor. Each convolutional filter contains only one value (which is fixed within the convolution), so the depth features are based only on the previous features of a point (Fig. 4a). Depending on the implementation, different numbers of convolutional layers and filters are used.

The third processing step is the aggregation of the features of the individual points into a global feature vector for the respective input block. This is done using the max-pooling function, in which only the largest value is kept for each feature (Fig. 4b). It results in a feature vector that can be used for classification of the point cloud block. For segmentation, this global feature vector is taken and appended to all individual point feature vectors. There is now a combination of inter-point and global features for each point, from which further depth features are generated. The depth features are used to

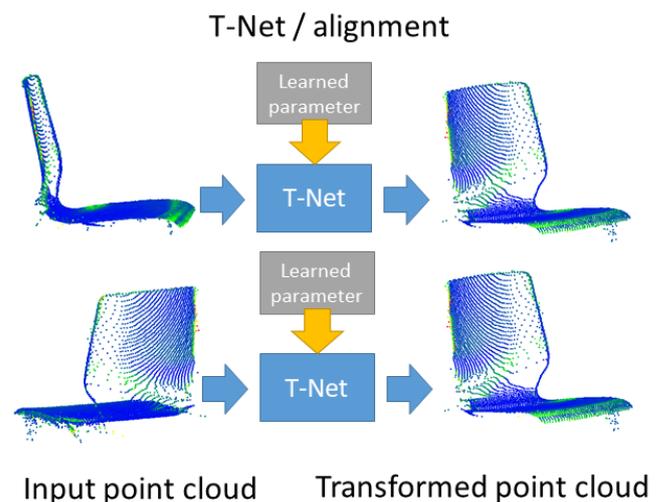


FIGURE 3. The intention of the T-Net module is that a point cloud is always aligned in a similar way by means of an affine transformation.

classify each point (e.g., with a softmax function) [13].

2) Limitations and advancements of PointNet

The central problem with PointNet is the selection of points for a block. This for instance is the case, if the area to be segmented semantically is very large, the point densities are in-homogeneous or different frequent classes are included. Regarding this challenge, different extensions, such as Point-

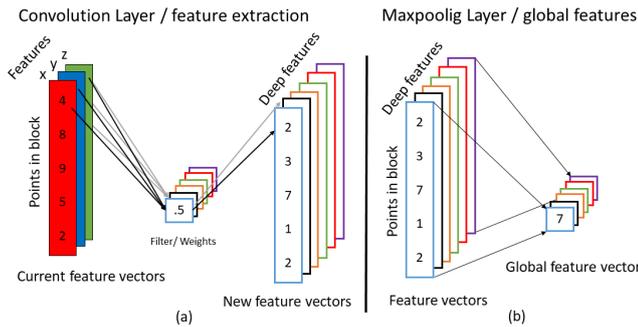


FIGURE 4. Convolution (a) and max-pooling (b) functions for feature enhancement and aggregation with PointNet.

Net++ [45] or a systematic neighborhood searches, such as by [46] have been developed. However, these developments also encounter limitations with extra-large and highly detailed datasets.

B. UNBALANCED CLASS DISTRIBUTION

One major concern for the semantic segmentation task solved with ML methods is, that different semantic classes in real-world data consist of different numbers of individual data objects [47]. For example, the background of an image is described by the majority of individual pixels and therefore it is learned more frequent by most ML algorithms. Often the algorithm learns only the background, because this way the highest accuracy is achieved for the whole dataset [31].

Basically, this problem exists for all ML methods, such as Support Vector Machine, k-Nearest-Neighbor (kNN), K-Mean Clustering, Convolutional Neural Network (CNN) and all kind of data types, such as data series, images, image databases or point clouds [48]. Various methods are developed to solve the class imbalance problem for certain data types. These methods can be clustered into four method groups (Fig. 5).

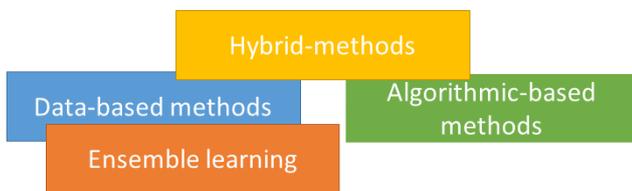


FIGURE 5. The four method groups to address the problem of unbalanced class distribution. Arranged according to similarities of methods.

The first method group, the data-based methods, encloses all methods, which actively change the number of the individual data objects (e.g., points or images). The dataset is filtered or augmented in such a way that the number of objects between different classes becomes equal or more similar.

Methods that reduce the number of data objects are referred as under-sampling (US) methods. These methods randomly [49] or systematically [50]–[52] select data objects per class to establish equality and ensures that only original (measured) data is used. The US methods have the central

disadvantage that parts of the knowledge are not used and therefor learning is only performed on a subset of the information. The use of only a subset could lead to changes in the local neighborhood [31].

Unlike US, random [49] or systematic [48], [53]–[55] over-sampling (OS) methods enlarge the dataset and augment it with artificial or duplicated data. One popular method is the Synthetic Minority Oversampling Technique (SMOTE) by [56], where the neighborhood is considered to control the OS method. The datasets become large without gaining any new knowledge. In addition, typical data objects in the infrequent classes are emphasized strongly, therefore the trained model may not transfer well to new unknown datasets. Methods that minimize the adverse influence of both approaches use the original and OS datasets in different training phases [57] or combine the OS and US methods, each based on the epoch result [48].

The second group of methods are the **algorithm-based methods**. They modify the learning algorithm and aim for a stronger impact of classes with fewer data objects. That can be either done by adapting the loss function [58]–[60], modifying the network architecture [61]–[64] or weighting the predictions [65], [66]. Learning can advantageously be done directly with raw data. But, if the class differences are very large, weighting can lead to a wrong relationship and a minor class may becomes too dominant.

The third group of methods are named as **hybrid methods**. They apply data- and algorithm-based methods together. The data are combined in a first phase at the level of ordinal features [47] or at the level of derived features to obtain highly differentiable features in the training data. The features can be created by grouping the initial features and deriving new features [67]. Other approaches create embedded features and adjust it in favor of the minor class [68] or taking into account possible high and low classification probabilities based on the feature distribution within the classes and its boundaries [69]. Hybrid methods are applied to CNN such that remaining differences due to the equalization of class sizes or optimization of the data are made by adjusting a loss function or using multiple loss functions.

The last method group is **ensemble learning**. These methods are applied to traditionally weak learning methods, such as Decision Trees or K-mean clustering. In ensemble learning, different classifiers or the same classifier are trained with different combinations of data or parameters. The results of all classifiers are evaluated to a combined result using hard or soft voting [70] (stacking and bagging). Boosting, as in *SMOTEBoost* [71], can be used as an alternative. Here, after each run, the classification parameters (e.g., selection of training data) are adjusted so that more attention is payed to hard-to-learn features.

C. HIERARCHICAL CLASS COMBINATION FOR SEMANTIC SEGMENTATION

The term *hierarchical semantic segmentation* is used in two definitions. The first definition is about the geometric size

change of the segments. The segments can grow (segments are merged) or shrink (segments are split) in the integrative and hierarchical segmentation process. The names and numbers of the classes always remain the same. In this approach, the semantics is added to the segments in a subsequent classification [72], [73]. Often, the point clouds are transformed into graphs, which are gradually refined or generate local features [74]–[76].

The second definition focuses on different classes at different stages of the segmentation. Here, the class definition is hierarchical and the semantic information changes by each level. This definition of hierarchical semantic segmentation is less described in literature, because for a semantic segmentation usually a fix set of semantic classes is defined in advanced and the process is done in one step. Frequent and infrequent classes are determined and segmented in the same step. In contrast, if the set of semantic classes is complex and/or oriented to a predefined hierarchical semantic schema, such as the CityGML [77], the Industry Foundation Classes (IFC) [18] or a non-institutional schema [78]–[80] another strategy can be applied. This performs semantic segmentation in several sub-steps. Semantic schemes usually have multiple aspects, such as geometry and semantic, and are organized into LoD [17]. The semantic LoD determines which class is determined in which level. Thereby, for each point only one class should be defined in one LoD. As shown in [81], this approach can help to better distinguish semantic classes with similar geometric features that appear in different LoDs. In addition, a combination of features from different LoD can help to increase the semantic accuracy for a semantic segmentation [81].

III. DATASET

Our dataset consists of more than 76 million individual points representing 27 rooms of the HafenCity University Hamburg main building (Figs. 6, 7 and 8). A subset of the point cloud was created for this work and contains the class Erroneous Points (subset A). This subset was extended by an existing dataset without the class Erroneous Points (subset B). Subset B was originally created for the Level 5 Indoor Navigation project [82] and was reorganized and improved for our experiments. The dataset is organized by rooms, which can be selected individually. The rooms are different in terms of furnishings, usage and shapes. Seminar rooms, lecture halls, offices, coffee kitchens, corridors and entrance halls are present in the dataset.

All rooms were surveyed using terrestrial laser scanners Z+F Imager 5010 or 5016. The survey was performed with a resolution of 6 mm at a distance of 10 m and the quality level "normal" [83]. Small and good observable rooms up to about 75 m² were surveyed from a single viewpoint. Larger or winding rooms were surveyed with multiple viewpoints so that all furnishings and building parts were captured completely. Small coverage gaps (e.g., on walls or on the floor due to obscuring furniture) are present in the data and accepted if the overall geometry of the semantic classes per

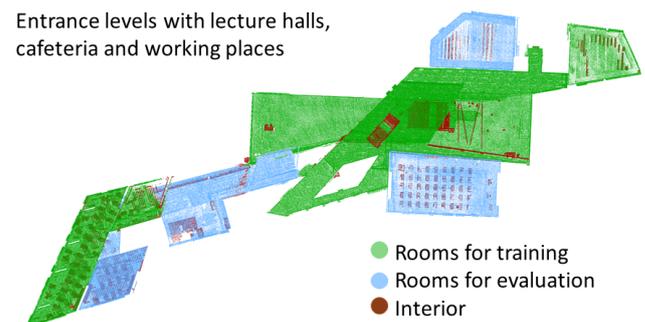


FIGURE 6. Point cloud dataset from the main building of HafenCity University Hamburg (entrance level).

room can be derived from the point cloud (Fig. 9).

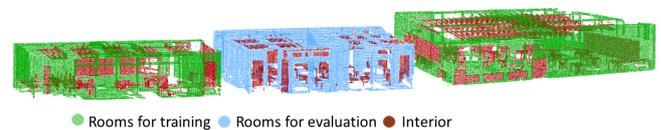


FIGURE 7. Point cloud dataset from the main building of HafenCity University Hamburg (office level).

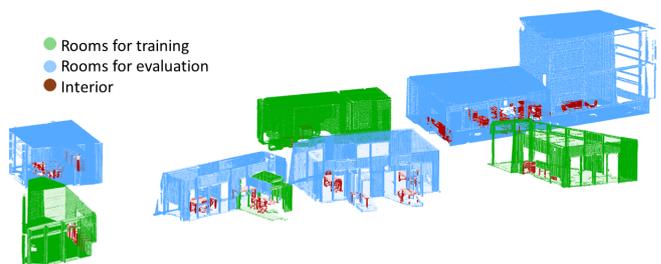


FIGURE 8. Point cloud dataset from the main building of HafenCity University Hamburg (lecture hall level).

The registration of the individual point clouds were carried out via discrete targets, which were measured automatically and manually in the scanned point clouds. Using the coordinates of a geodetic net measurement (via total station) the scanned point clouds are transferred into a global and uniform coordinate system (geo-referencing). The division by rooms was done in a manual segmentation procedure. For this purpose, the spaces are roughly selected in the entire point cloud and a partial point cloud is copied. The partial point clouds are processed so that only points of the respective room are included. This procedure leads to a more complete point cloud, because points of viewpoints in neighboring rooms are considered.

The second segmentation step is based on the semantic classes and was performed with CloudCompare [84] and Autocad Recap [85]. To achieve a high quality of the manually classified points, each point cloud was semantically segmented at least three times by different annotators. The annotators were previously trained in the task and received feedback on intermediate results. The individual segmen-

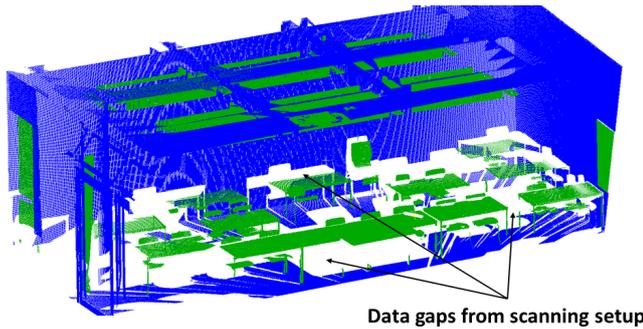


FIGURE 9. Gaps (white areas) in the point cloud caused by occlusions (e.g., furniture), and which are tolerated in the dataset.

tations of the same rooms were combined so that coarse individual errors are removed.

IV. METHODOLOGY

The influence of certain DHPs, especially for point clouds are still little systematically studied. The semantic classes are usually defined according to the application, such as processing areal LIDAR point clouds for industrial use [2] or having a Scan2BIM application [4]. The semantic segmentation of all defined classes is usually achieved in one step. [86] observed that the class definition and the class content have an influence on the semantic segmentation result. As an alternative to the application-oriented class definition, an algorithm-oriented class definition is also possible. This insight leads to a process in which the DHPs are set in favor to the algorithm by considering: The number of classes, the number of points per class, the presence or absence of erroneous points and the geometric difference of the objects in different classes. To investigate the DHP, an application and experimentation environment (AEE) was developed in which the common HPs and DHPs can be easily customized. The AEE offers different options for the point cloud augmentation using balancing (improvement) techniques.

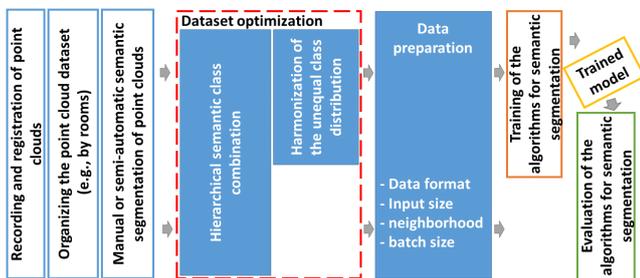


FIGURE 10. Processing steps for semantic segmentation of point clouds. The filled boxes are examined in detail.

The investigations follow the workflow shown in Fig.10. The data recording is followed by the registration, the organization of the sub-scans, and the manual semantic segmentation of the point clouds, so that these can be used as training and evaluation data. These steps are followed by a dataset optimization, which is the central focus of this work

(Sections IV-B and IV-C). Next, the data is prepared for the processing step with the chosen automatic semantic segmentation method (Section IV-A) and the algorithm is trained. The efficiency of the training is evaluated with "unknown data" of the same dataset (e.g., other rooms).

A. APPLICATION AND EXPERIMENTATION ENVIRONMENT

Our work is based on the DL-architecture PointNet [13], for which optimal HPs were determined based on comprehensive preliminary investigations and literature research [41], [87]. PointNet is one of the established and foundational DL-architectures which makes our investigation results comparable with other studies. All parameters of the network architecture (except for the number of classes) remain as in the implementation of [88]. The AEE is developed that PointNet can be replaced by other point-based DL-architectures.

The main drawback of PointNet is that only a small number of points and only local features are used to assign a point to a class. Our approach to control the input of points is simple and is based on a random and uniform splitting of the point cloud into three equally sized sub-point clouds and the determination of a Local Neighborhood Box (LNB). The origin of coordinates of the entire point cloud is defined by the smallest values for the x- and y-coordinates. This origin of coordinates is used for the first sub-point cloud. For the following sub-point clouds, it is shifted in the x-y-plane by a fraction of the LNB edge length and additionally rotated by a fix angle (Fig. 11). For each of the shifted and rotated sub point clouds, the LNB are determined using the structure algorithms of *pyntcloud* library.

The local neighborhood is defined by a 1 x 1 m LNB whose height is the maximum possible room height of the dataset. By shifting and rotating, six different local neighborhoods are created for each original LNB. The rotated point cloud is an extension of the original point cloud. From each LNB a certain number of n randomly selected points is taken as network-input until all points have been fed into the network. If there are not n points left, the input is filled by random copied points from the LNB. In addition to the global normalized room coordinates (x_{glo} , y_{glo} , and z_{glo}) and the point normals (x_n , y_n , z_n), the local normalized coordinates of the LNBs (x_{loc} , y_{loc} , z_{loc}) are calculated. These nine geometric features are used as input features for all experiments.

B. METHODS FOR HARMONIZING THE UNBALANCED CLASS DISTRIBUTION

The semantic classes of point clouds from real objects differ by the number of points. Objects, such as walls and floors, take up more area (as well as points), compared to objects, such as doors and erroneous points. This is due to the fact that most surveying systems regularly scan surfaces with a fixed angular increment related to the sensor, which changes with distance. In addition, the measuring systems capture areas and not edges. Two backwards arise from the capture conditions for the training of semantic segmentation meth-

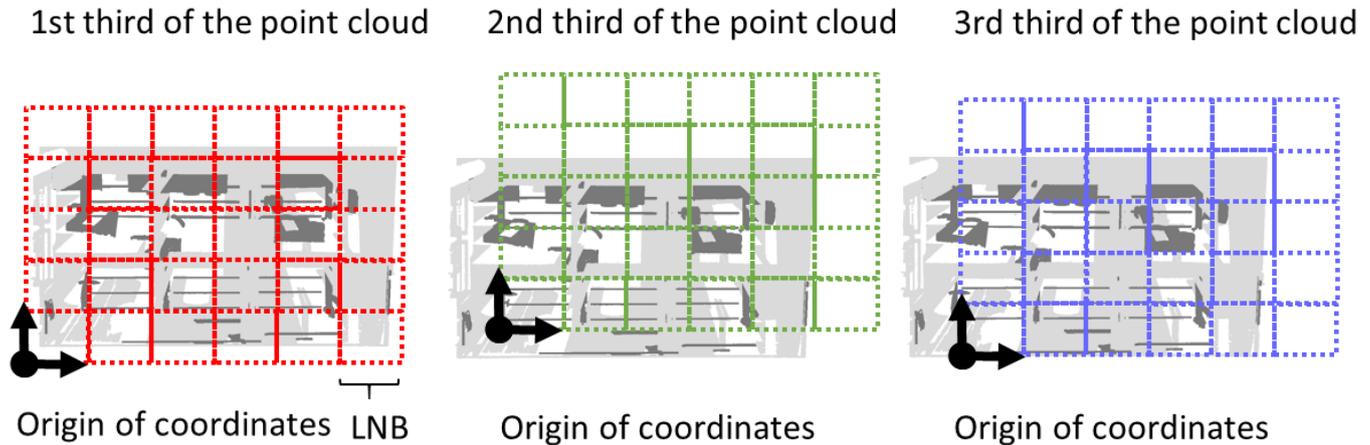


FIGURE 11. Describing the neighborhood for PointNet inputs using overlapping LNBs.

ods. First, a lot of information is collected which provides no or little new information for the separation of semantic objects. Second, there is often a lack of information about geometrically complex and variable objects, as well as of the class edge areas.

ML-based semantic segmentation methods learn a relationship between input features and semantic class over large amounts of data and try to determine an optimal separation over the majority of point features. If one class is dominant in the number of points, it can be observed that the best results are obtained by assigning almost all or all points to this class. In processing of medical images, this problem is well known by the fact that only a few pixels show the anomaly and most pixels show normal organs [89]. For our point clouds, the problem is transferable, because most points belong to frequent classes, such as wall, floor or ceiling. The underlying idea to solve this topic is to focus on the information that is important for the separation of the classes and to increase its importance. These is usually done by augmenting the points of the infrequent classes. The emphasis on the infrequent class(es) is investigated in the experimental studies of Section V-D by means of four techniques. These techniques are the SMOTE [56], the stack augmentation (SA) and two adaptations of the loss function.

1) SMOTE

In the applied implementation of SMOTE, the amount of all classes are expanded to the number of points of the largest class. Thereby, all classes consisted of the same number of points and are given homogeneously distributed into the model. Other variants of the SMOTE implementation, e.g. up to 50% of the size class or a combination with a US method could be examined alternatively. By using SMOTE, the expansion is controlled by the local neighborhood, so the later learning focus is placed on the areas of the point cloud that describe infrequent and usually more complex object classes. SMOTE uses the kNN algorithm to determine the k nearest neighbors of each point. The number of neighbors

k is the factor by which the point cloud is augmented. If $k = 1$, then the point cloud is doubled. If the point cloud should be augmented to a certain number, then the multiplication number is $k + 1$. The unnecessary points have to be (randomly) deleted afterwards. For the calculation of the coordinates of the augmented points, the vector between the starting point and the nearest point is determined. The vector between this points is multiplied with a random value from 0 to 1 and added to the starting point. The coordinates of a new point are located in between both original points (Fig. 12). With SMOTE the density of the point cloud is artificially increased in the areas of the minority classes [56].

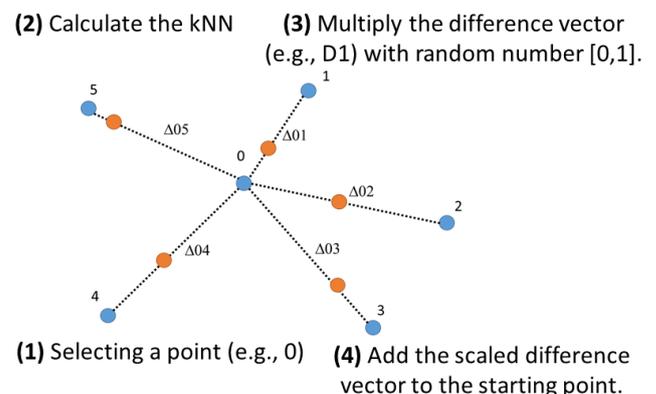


FIGURE 12. Calculations for data augmentation with the SMOTE method by [56]. In this example the point cloud is multiplied by $k = 5$ times.

2) Stack augmentation

SA is also a data-based augmentation method. However, the data is not augmented in a process ahead of DL-method and not to a fix amount of points. Instead the dataset is expanded during the creation of the training dataset. The advantage of the SA is a smaller increase of data, thus the augmentation is mainly applied to points of infrequent classes. The basic idea of the approach has been developed by [55]. They split the point cloud into chunks as network-input (similar to a

voxel). Within a chunk the number of points was reduced to a fixed amount of 4096 points. The content of the chunks is analyzed in regard to the number of points per class. Chunks with many points of infrequent classes are augmented more frequently than chunks with many points of frequent classes. The frequency of each chunk is determined by a nonlinear function. Using this data augmentation strategy, [55] are able to achieve an increase of about 10% for recall and precision for the outdoor laser scanner dataset *Semantic3D* [37].

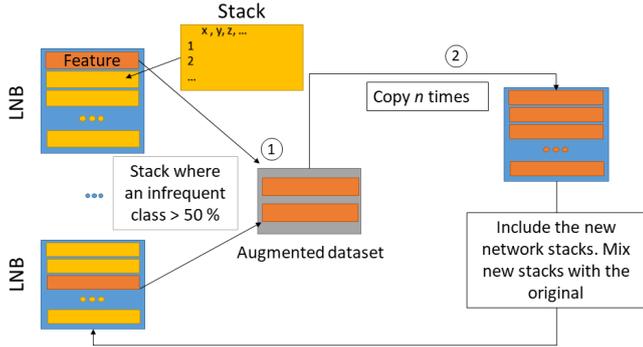


FIGURE 13. Process of stack augmentation for an optimization of the class distribution.

We adapt the method of [55] for our data processing and simplify the calculation for the augmentation factor. The augmentation and the analysis was performed on the basis of a stack with 1024 points, which is the input for the PointNet. Stacks are similar to chunks, however they do not have a fixed spatial dimension, since a stack consists of randomly selected points of an LNB (Fig. 13). The augmentation degree is determined by calculating the target proportion for each class (if all classes would be equal) and comparing it with the actual distribution. If the actual proportion of a class is smaller than the target proportion, then stacks in which this class is dominant are copied into an augmentation dataset (Fig. 13, step 1). The augmented dataset is duplicated after all stacks have been analyzed. The number of augmentations (n) is determined by the fact that the smallest class must have its target proportion (Fig. 13, step 2). For instance, the points of a point cloud should be classified into three semantic classes, so the target proportion is 33.3% to which the smallest class is augmented.

With this augmentation method, the focus should be directed to the infrequent objects in the point cloud but without losing information of large objects. Especially points in the edge zones, where small and large semantic objects meet, should be used more in the training. Stacks that contain a majority of infrequent classes should be augmented. This can be a stack, that consists only of points of the infrequent classes (Fig. 14b), but also stacks which contain few points of the frequent classes (Fig. 14a). Stacks with a majority of frequent classes are not augmented (Fig. 14c).

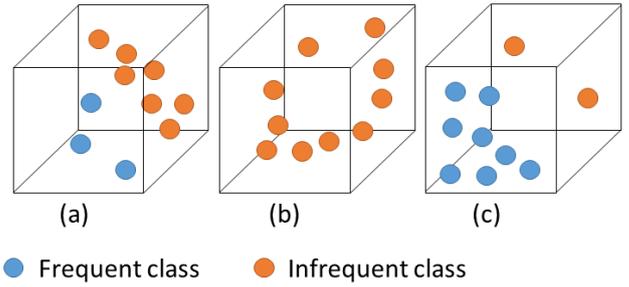


FIGURE 14. Geometric visualization of the stacks for input to a network. The black box represents the boundaries of a stack. (a) Majority of points is from the infrequent class. (b) Only points from the infrequent class are present. (c) Majority of points are from the frequent class. This stack will not be used for augmentation.

3) Weighted loss function

The third and fourth methods for minimizing the unbalance class distribution are algorithm-based and addresses the loss function that is used to calculate the classification error after each training pass. The loss function type used in this work is the Categorical-Cross-Entropy (CCE) loss function which is extended by two weighting options. The concept of loss calculation is shown in Fig. 15 and can be briefly described as follows.

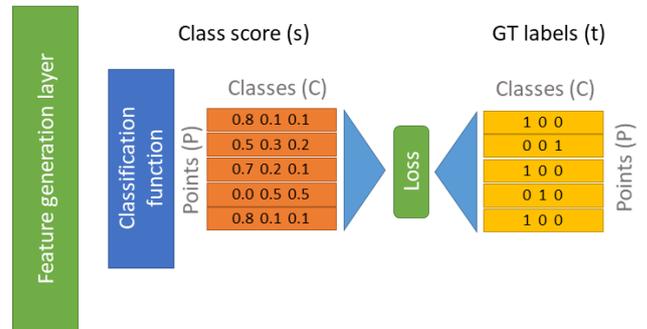


FIGURE 15. Process of feature extraction, classification and loss calculation at PointNet. Prediction class score (s) and ground truth label target (t) vector as one-hot encrypted matrix.

The raw training data given to the network is unbalanced, and the depth features are computed based on the original data. Applying a classification function (e.g., softmax), a one-hot-encode class vector for each point is determined based on the features. The class vector consists of the same number of elements as possible target classes (C) exists. In the case of the softmax function, the vector is normalized such that the vector sum is always one and the values of the vector express the probability for each class. In the predictions of the DL-method, commonly the maximum value of each point vector is determined and the one-hot encryption is decrypted. Also, the class vector is used to determine the loss during the training. For PointNet the CCE function from (1) is commonly used.

$$CCE = - \sum_{c=1}^C t_c * \ln(s_c) \quad (1)$$

The CCE function is used to calculate the loss of a classification by summation of all multiplications between the logarithmized elements of the class vector (s_c) and the corresponding elements of the target vector (t_c) from ground truth (GT) label. Thereby, mean loss of each input stack and for the entire point cloud is determined. The mean loss does not distinguish whether the classes of the points are difficult or easy to learn or if the points are frequent or infrequent. Classes that occur infrequently and have a high loss are included in the mean value to a less extent than classes that occur frequently. The algorithm learns frequently occurring classes better. To minimize this disadvantage of the infrequent classes, the CCE can be improved by a weight vector (w), as described in (2).

$$WCCE = - \sum_{c=1}^C t_c * w_c * \ln(s_c) \quad (2)$$

The target vector (t_c) is multiplied with the weight vector (w_c), allowing the loss of the infrequent classes being emphasized in the mean loss. This loss function is called weighted CCE (WCCE) loss function and is shown in (2).

$$w_c = 1 - \frac{P_c}{P} \quad (3)$$

The calculation of these weights is usually done by the class distribution [90]. The weights in our experiments are calculated and tested using two independent experiments. In the first experiment, the proportion of a class is determined by calculating the ratio of the amount of points of one class (P_c) and calculating the amount of all points (P). The ratio of P_c and P boost the frequent classes, so it must be subtracted from 1 to emphasize the infrequent classes (3). This method reduces the loss, which can lead to a too early termination of the training phase. To minimize this reduction of the loss, the weights can be calculated according to (4).

$$w_c = \frac{1}{C} - \frac{P_c}{P} + 1 \quad (4)$$

The minor or superior proportion of each class is calculated by (4). Minor or superior proportion result from the difference to a class distribution having classes of the identical size. This leads to the fact that frequent classes get weak weights and infrequent classes get strong weights without changing the total amount of the loss.

The two WCCE functions are developed on the basis of [91] code and have been integrated as an option into the AEE. A major advantage of this method is that the weighting is only effective during training and (theoretically) the algorithm does not have to be trained again on the original data. The feature extraction and the classification itself are only indirectly influenced by learnable weights.

C. HIERARCHICAL SEMANTIC CLASS COMBINATION

The class definition specifies the semantic classes in which a point should be subdivided. The size of the individual semantic classes is indirectly given by this class definition. In applications where weak ML methods, such as Random Forrest, are used, hierarchical class definitions are used to increase the efficiency [92], [93]. A hierarchical class definition consists of several levels. General classes are defined in the top layer, which are subdivided further and further until the target classes for an application are reached. For instance in the top layer, building parts and interior can be distinguished, which can be further distinguished into classes, such as Wall, Floor, Ceiling or Window. Using a hierarchical class definition can be beneficial for the semantic segmentation because fewer distinctions in one step need to be made and the imbalance of the classes are minimized by a optimal definition. The study of [81] on PointNet++ shows that combining feature vectors from different hierarchical layers of the class definition results in a better discrimination for some classes. In their research unmanned aerial vehicle LIDAR data is analyzed and [81] state that many different semantic classes are geometrically similar. If these geometric classes are already separated by previous levels, confusion between these classes is eliminated.

Based on the work of [81] and our theoretical considerations, we developed a class definition for indoor applications, which is summarized in Fig. 16. The full hierarchical class definition is shown in Tables 13 and 14 in the appendix. By developing this, trade-offs were made between semantic reasonableness, the different geometric shapes of objects in a class, and class sizes. The goal is to form classes that are usable for a possible semantic application, that are geometrically different, and are as similar in distribution as possible. In particular, the equal class distribution is often in contradiction to other goals. These goals can possibly be achieved by combining the infrequent and geometrically similar classes Door and Windows into Opening.

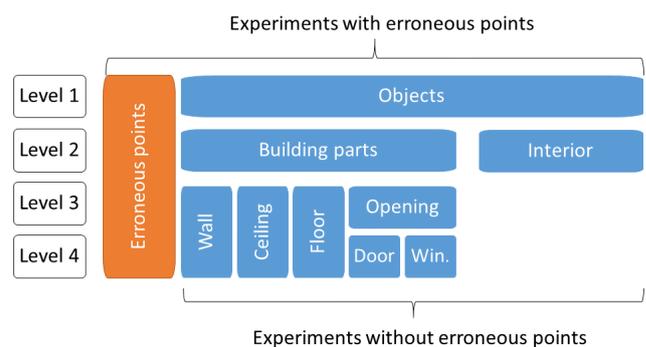


FIGURE 16. Semantic model for the examinations. A distinction is made between the main classes of Building parts, Interior and Erroneous Points. Sub-classes are considered separately starting from level 2. Each level can and cannot include erroneous points.

In addition to the classes for real objects, the class Erroneous Points is formed as an extra semantic class for a subset of the dataset. This semantic class includes the points that are

caused by the measurement system, the measurement setup or unfavorable object properties (e.g., highly reflective). The works from [26] and [94] state that this class can have a measurable influence on the semantic segmentation results. Usually the class Erroneous points is determined with a correctness and precision of less than 20%. Even if the erroneous points are difficult to determine, such a semantic class can theoretically contribute to an improvement of the other classes [42]. In the following experiments, semantic segmentation is performed with and without this class. It should be noted that in the case of semantic segmentation without the class Erroneous Points, these points were removed from the point cloud using parameter-based filters and manually segmentation.

The semantic class definition is structured in such a way that only a subset of the points is segmented semantically in the lower levels. Thus, it is assumed that the previous level already has sufficient segmentation accuracy. In our experiments, the manually created semantic point clouds are used, so that a consideration of the maximum possible segmentation is performed. For the investigations, different splits resp. semantic generalization degrees are used in the 3rd and 4th level and all investigations were carried out for all augmentation methods of Section IV-B. The classes are split into seven combinations, the classes for each combination are shown in Table 1.

TABLE 1. Used class combinations including sub-classes.

Combination	Classes
1	Erroneous Points, Objects
2-1	Erroneous Points, Building parts, Interior
2-2	Building parts, Interior
3-1	Erroneous Points, Floor, Ceiling, Window, Door, Wall
3-2	Floor, Ceiling, Window, Door, Wall
3-3	Floor, Ceiling, Opening, Wall
3-4	Erroneous Points, Floor, Ceiling, Opening, Wall

V. EXPERIMENT SETUP

In this work, the AEE is used for analyzing the influence of the DHPs. The four data and algorithm-based augmentation methods for minimizing the influence of class size differences, as described in Section IV, are investigated in detail. In addition, the influence of a step-wise semantic class definitions is determined.

A. RESEARCH FIELD AND QUESTIONS

The semantic segmentation of point clouds makes point clouds interpretable for machines. It is one of the key steps for the automated high-accurate digitization of the real world, as performed by surveyors. From a surveyor's perspective, the data, the data quality and the DHPs are of high interest for the evaluation of different point clouds in terms of reliability, efficiency and accuracy. For the development of an automatic processing method, it is necessary to estimate the influence of the individual DHPs. The DHPs, class combination and class distribution are examined in detail in the following, in order

to determine these influencing variables for the following developments or to neglect them, if they do not show a significant influence. Our investigations clarify which improvement for the semantic and geometrical accuracy can be achieved with a data or algorithm-based data augmentation method. Furthermore, we investigate if classes can be learned better by a step-wise segmentation of the point cloud.

B. HARDWARE, SOFTWARE AND HYPERPARAMETER

Training for all experiments was performed on a single workstation. The parameters of the hardware used for the computations are summarized in Table 2. The AEE was developed entirely in Python and uses Tensorflow and Keras as DL-frameworks (Table 3). Programming was performed for a single GPU. An adaptation for a multi-GPU system is given.

TABLE 2. Hardware used for our AEE development and in the experiments.

CPU	GPU	GPU RAM	RAM
AMD Ryzen Threadripper 2970WX	GeForce RTX 2080 Ti	11 GB	64 GB

TABLE 3. Software and software versions used for our AEE development and in the experiments.

DL-Framework	Program Language	GPU Accelerator
Tensorflow 2.3.0	Python 3.8	CUDA 10.1

All experiments on one class combination with the different data augmentation methods are performed as one block of experiments. In order to compare the methods, the semantic segmentation with the original point clouds are performed for each level. The duration of the training varied from 49 to 287 minutes due to the size of the given subset and the data expansion methods. As an example, run times for all basic types of class sets are shown in Fig. 15. Longest run times were observed for the SMOTE and the SA method, since the number of points increases for both methods. A reduction of the run time was observed for the WCCEa method for the predominant cases, which can be explained by the general reduction of the loss (Fig. 17).

The initial set of common HPs were determined based on the work of [1], [2], [8], [13] and optimized empirically. The optimized HPs are summarized in Table 4. To reduce the learning time, early-stopping was introduced. The training is stopped after 25 epochs in which the metric *eval-loss* does not decrease by more than 0.01. To optimize loss, the common *Adam* optimizer [95] is used with a learning rate that is reduced while training progresses. This should help to increase the learning efficiency. Batch size, number of epochs, points per stack and stack size were selected identically for all experiments.

C. EVALUATION PARAMETER

The evaluation of the semantic segmentation is carried out with the three evaluation parameters recall (RP), precision

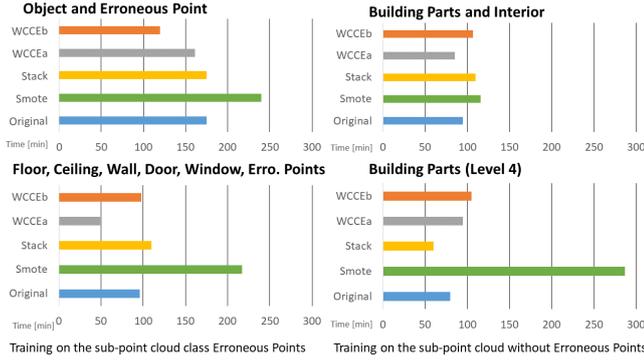


FIGURE 17. Selection of training run times for the class combinations and the different data augmentation methods.

TABLE 4. Selected HPs for all experiments.

Stack size 1024 points	Batch size 16	Epochs 1000	Early stopping after 20 Epochs
No. of features 9	Lear. rate 0.001 to 0.00025	Stack dim. 1 x 1 x 6 m	Indep. trainings 9

(PP) and standard deviation of the false positive (SDFP) points. The parameters *True Positive* (TP), *False Negative* (FN) and *False Positive* (FP) points are determined by comparison with the GT labels. The parameter n_C in (7) (geometric accuracy) stands for number of points for the current class. x_i is a predicted point for this class and x_{GT} is the closest point to x_i from the set of GT points for this class (reference point cloud). The main evaluation parameters are determined using (5) to (7). These scores were determined at the room level and were averaged for the analysis.

$$RP = \frac{TP}{TP + FN} \quad (5)$$

$$PP = \frac{TP}{TP + FP} \quad (6)$$

$$SDFP = \sqrt{\frac{1}{n_C} \sum_{i=1}^{n_C} (x_i - x_{GT})^2} \quad (7)$$

The SDFP points describes how precise the geometry of an object class is and can be seen as a supplementary parameter to the semantic PP. If SDFP points is greater than an application-related threshold (e.g., 100 mm), then gross segmentation errors are present. In most cases the segments of this object class cannot be used for the target application. The geometry of the segments is strongly changed (enlarged). If SDFP is smaller than the threshold, this parameter can be used to examine whether the segments are suitable for a particular LoD representation. This evaluation parameter can vary between different classes in a dataset.

Class equality (CE) for class combinations is introduced as an additional parameter and is determined using (8). Values close to 1 indicate an unequal class size and values close to 0 indicate an equal class size. The parameter CT_c is the

proportion in case of an equal distribution of points per class and the parameter CA_c is the actual proportion of points per class.

$$CE = \sum_{c=1}^{P_c} \|(CT_c - CA_c)\| \quad (8)$$

A detailed class definition is available for each class combination. Further parameters concerning the point cloud quality were not considered for the analysis of these experiments. The manual semantic segmentation is reviewed for major errors.

D. PROCEDURE OF THE EXPERIMENTS

The experiment can be divided into two phases as shown in Fig. 18. In phase 1 of the experiment, 35 different experiments consisting of data augmentation methods and class combinations are studied. All experiments were initialized with random learnable parameters (weights). The weights of the network are randomly but they were used identically for all experiments, so only the influence of the training process is shown by different segmentation performances.

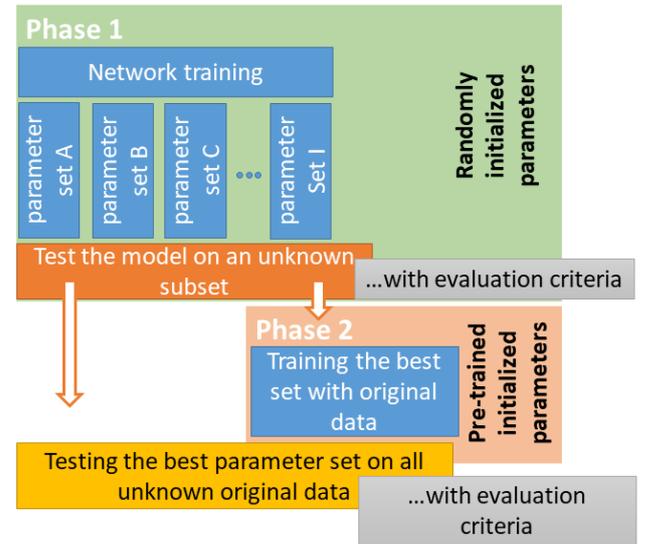


FIGURE 18. Evaluation and Transfer Learning strategy.

An analysis of the results is performed according to the scoring scheme of Fig. 19. The best weight set is used for a detailed investigation and the Transfer Learning (TL) in the second phase. In the TL phase, on the basis of the best weight set, nine new trainings are executed per combination without using a data augmentation method. The data augmentation methods SMOTE and SA change the point clouds and new unwanted patterns may appear. These patterns can adversely affect the generalization, so they should be avoided if possible. The aim of phase 2 is to describe the influence of TL.

The evaluation scheme defines that at least half of the points in each class are correctly identified and that there

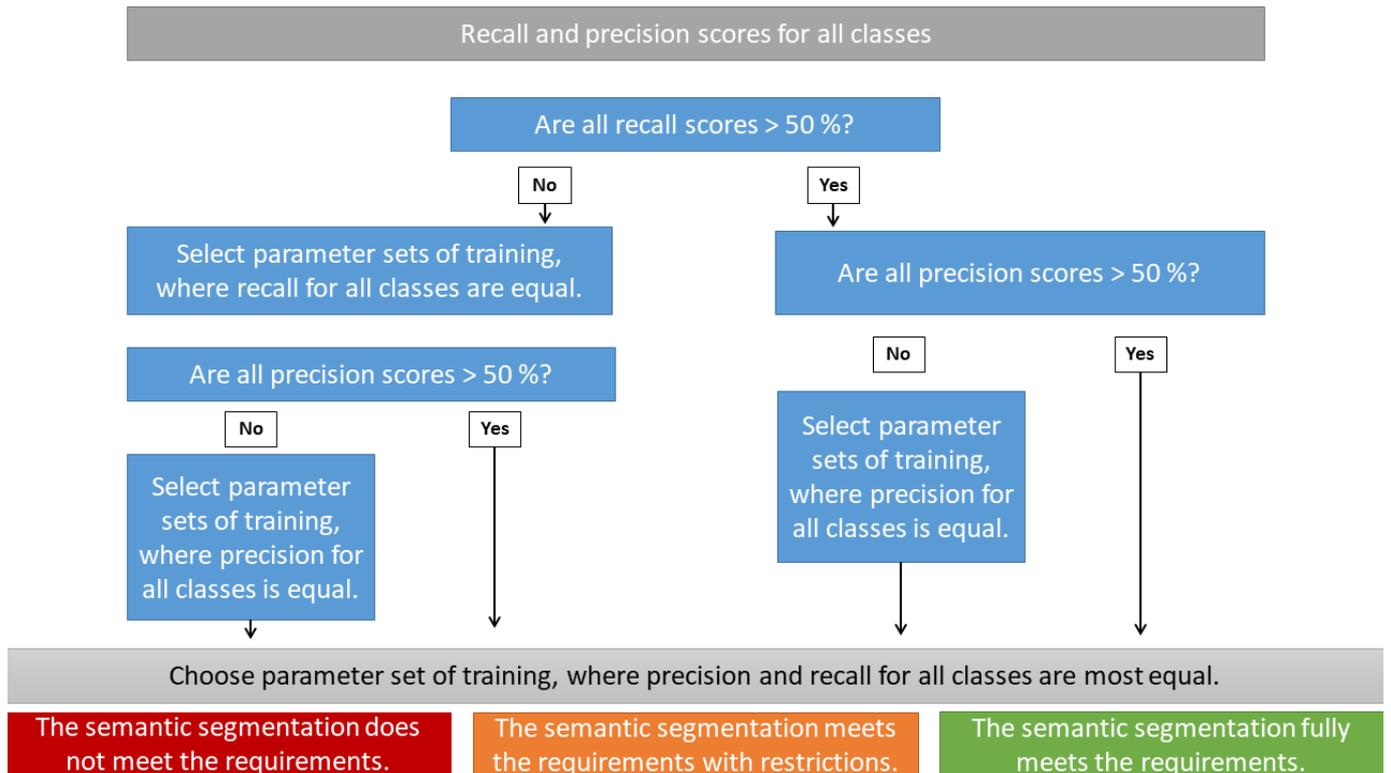


FIGURE 19. Selection scheme for Transfer Learning and semantic segmentation evaluation.

are not more false points than true points in the class ("Requirement fulfilled"). This requirement seems logical from a human perspective, assuming that something is learned, if it is done more frequently correct than incorrect. However, in semantic segmentation with DL, this requirement is rarely met, and for a large number of tasks it does not have to be met. In order to evaluate which weight set provides the best performance, two additional levels have to be defined. The levels: "Requirements meet with restriction" and "Requirements not fulfilled". The requirements are met with restrictions, if either RP or PP is less than 50% for one class. The requirements are not fulfilled, if both RP and PP is less than 50% for one class. In these cases, the best weight set is the one with the highest detectability and PP of the weakest classes. This evaluation scheme is primarily used to determine the best weight set for the subsequent comparison of the combinations.

VI. RESULTS AND DISCUSSION

Training results are stored as weight sets and can be loaded for evaluation with the full dataset. The evaluation of the weight sets from phase 1 and 2 is performed with the full test dataset or subset B, as described in Section III. From nine weight sets per combination, the weight set that preforms best to the evaluation scheme of Fig. 19 is selected and is analyzed in more detail below. In addition to the four data augmentation methods (Section IV-B), a not adapted version of the network architecture (Base method) is trained for each class combination.

The influence of the investigated HPs is expressed by the evaluation metrics RP, PP and SDFP points for each class (Section V-C) and in the form of a class average. These evaluation metrics allow a detailed evaluation for the creation of a BIM, based on a semantically segmented point cloud. The RP shows how complete a class is detected and the PP shows how many points of other classes are erroneously assigned to the considered class. For the creation of a structural model (walls, ceilings and floors), it is important that the segments of the relevant classes are as semantically precise as possible (high PP) and the predicted segments are geometrically identical to the GT segments (low SDFP of points). A complete assignment of all points can often be considered as less meaningful for this application. However, a high RP for the class Erroneous Points is very important, since all erroneous points should be removed from the data.

A. CLASS COMBINATION 1

The first examined class combination (**combination 1**) consists of the two classes Erroneous Points and Objects. According to the test procedure (Fig. 18), three (intermediate) results are available for each class combination and each augmentation method. For the SMOTE method of combination 1 (shown in Fig. 20a - d), the following evaluations are based on the best-retrained (Fig. 20c) and the TL (Fig. 20d) semantic-segmented point clouds. The Base method and the SA method do not meet the requirements. All other methods meet the requirements with restrictions from the evaluation scheme. The CE rate is high for all methods, with 0.96,

except of SMOTE. The present class combination is unfavorable for DL-based semantic segmentation.

TABLE 5. Semantic accuracy of the class combination 1 (subset A). The symbols \uparrow and \downarrow indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.96	0.00	0.96	0.96	0.96
Precision					
Err. Points	47%	\downarrow 23%	43%	\downarrow 25%	39%
Objects	96%	98%	96%	99%	97%
Class average	71%	\downarrow 61%	\uparrow 96%	62%	68%
Recall					
Err. Points	44%	\uparrow 75%	42%	\uparrow 84%	\uparrow 59%
Objects	96%	\downarrow 82%	96%	\downarrow 82%	93%
Class average	70%	78%	69%	\uparrow 83%	76%

In Table 5 it can be seen, that the methods SMOTE, WCCEa and WCCEb increase the recognizability of the small class. The erroneous points are better recognized, which leads to a more precise class Object in this binary-class case. Less precise is the class Erroneous Points for these methods and more points of the class Object are recognized as erroneous points. For the class Erroneous Points the PP decreases by 23% (Table 5). The SDFP points improve for the methods SMOTE and WCCEa by approximately 10 mm (Fig. 21). The SDFP points for the class Object is smaller than 100 mm, so that erroneous points change the object geometry at most by this amount. A model of captured structure can be created with this uncertainty. Such a model can be used for indoor pedestrian navigation or creating a rough spatial map [96].

Applying TL in phase 2, the semantic and geometric accuracy of all methods are equal to the Base method. The TL does not provide any advantage in this case. The methods SMOTE, WCCEa and WCCEb without TL improve the separation of the classes. This can be seen for SMOTE by comparing Fig. 20a with Figs. 20c and 20d.

B. CLASS COMBINATIONS 2-1 AND 2-2

The second class combination consists of the classes Interior and Building Parts, with (combination 2-1) and without (combination 2-2) the class Erroneous Points.

For **combination 2-1** (Table 6), the Base method and the SA method do not meet the requirements. All other methods meet the requirements with restrictions. These findings are similar to combination 1. The CE rate for the Base, WCCEa and WCCEb methods is high with 0.76. The SMOTE method has the optimal class distribution and the SA method improves the rate to a moderate score of 0.54. The proportion of the smallest class (Erroneous Points) remains at 4% as in combination 1.

The larger classes Interior and Building Parts show a PP value higher than 80% and RP of higher than 65% (Table 6). The RP for these classes is decreased by the augmentation methods in favor of an increase of the erroneous points by up to 31% (e.g., for SMOTE). The RP for erroneous points of the method SA increases by 6% in comparison to the Base method at the lowest. However, the SA method is the only

TABLE 6. Semantic accuracy of the class combination 2-1 (subset A). The symbols \uparrow and \downarrow indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.76	0.00	0.54	0.76	0.76
Precision					
Interior	84%	89%	89%	88%	80%
Err. Points	44%	\downarrow 29%	46%	40%	\downarrow 34%
Build. Parts	94%	93%	93%	93%	95%
Class average	74%	70%	76%	73%	70%
Recall					
Interior	85%	\downarrow 65%	81%	78%	84%
Err. Points	42%	\uparrow 73%	48%	51%	50%
Build. Parts	94%	90%	97%	95%	88%
Class average	73%	76%	75%	75%	74%

method with a PP higher than the Base method for all object classes and the average SDFP points is lower with 202 mm (Fig. 22). Therefore, this method achieves the highest accuracy (Table 6 and Fig. 22). The PP of the erroneous points for all other augmentation methods decrease by a maximum of 15% (e.g., for SMOTE) compared to the Base method. In comparison, the object classes have a high PP of more than 80%.

The geometric accuracy varies with a SDFP points from 129 mm to 410 mm. These SDFP points are very high and indicate major errors in the segmentation as shown in Fig. 23. Interior objects located within a range of about 200 mm from the wall cannot be reliably recognized. In the further course of the investigation, it is shown that the ceiling and floor are better separable from the furniture. A separation of wall and interior is only possible with this very high inaccuracy. The lowest SDFP points for the building parts can be obtained with the WCCEb method. For an overview model of a building, the point cloud of the class building parts can be used. This point cloud can also be used as the basis for a fast manual or parametric algorithm-based further processing.

The TL of the augmentation methods with the Base method leads overall to a small improvement of the semantic accuracy for the object class, but disfavors the class Erroneous Points by a decrease in RP.

The investigated methods lead to an improvement in the detectability of the class Erroneous Points. The recognition and PP of the object classes are not improved by the augmentation methods. The semantic PP of these classes are high as shown in Table 6. The geometric accuracy is low by LoA1 (according to the schema of [16]) and as shown in Fig. 22.

For **combination 2-2** (subset B), all augmentation methods meet the requirements with restrictions. The CE rate for the Base, WCCEa and WCCEb methods is moderate with 0.46. The SMOTE and the SA method have an optimal class distribution (Table 7). No erroneous points are included in this dataset.

RP and PP of the Base method and the augmentation methods are at the same accuracy level. For the class Building Parts, the RP of the methods SMOTE, SA and WCCEa is reduced by up to 3% in comparison to the Base method. The recognizability of the class Interior is increased by up

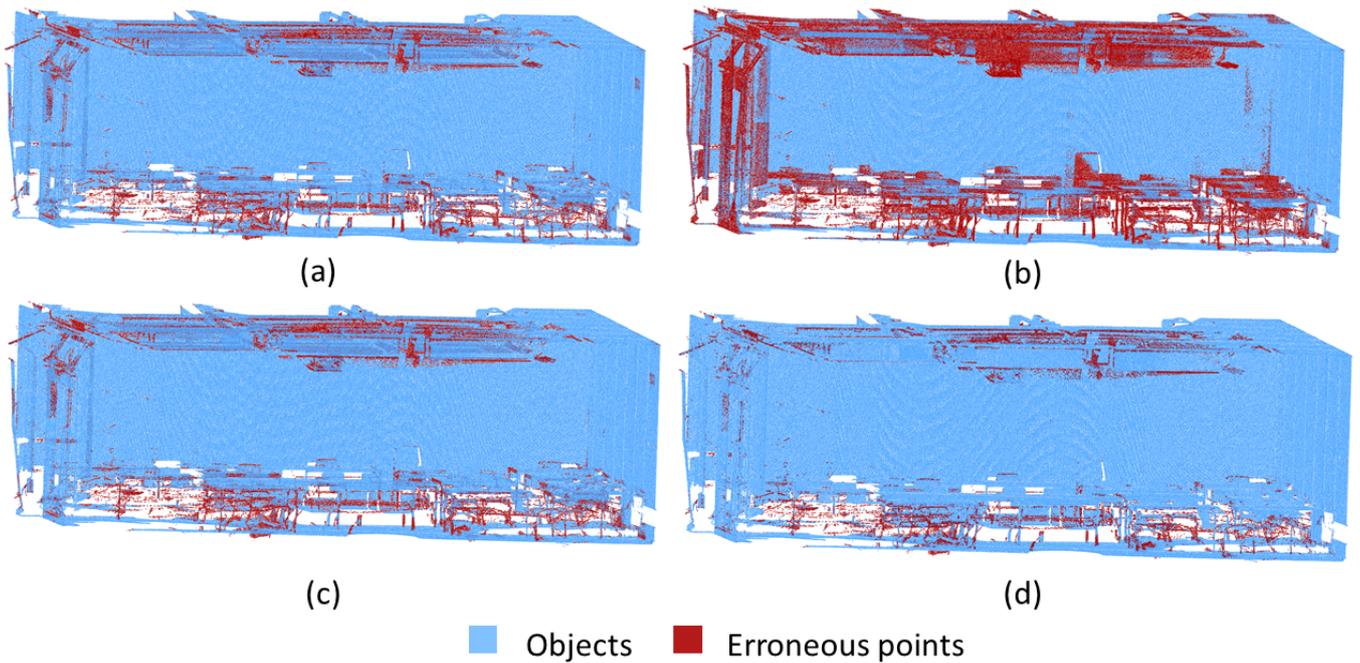


FIGURE 20. Sample point cloud of combination 1 with applied SMOTE method. (a) GT Point cloud. (b) Semantically segmented point cloud with random training. (c) Segmented point cloud by retrained best random version. (d) Semantically segmented point cloud with applied TL.

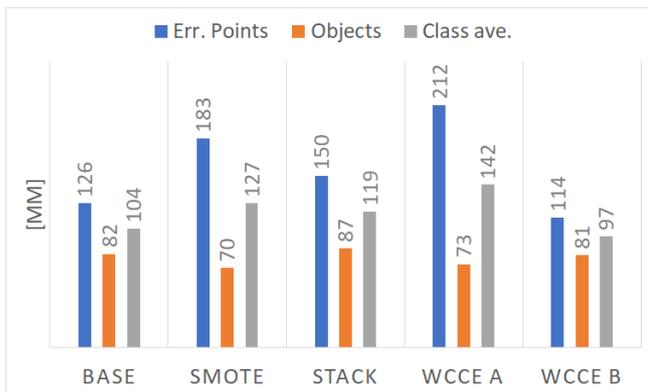


FIGURE 21. Geometric accuracy of the class combination 1 (subset A). The geometric accuracy is expressed by the SDFP points.

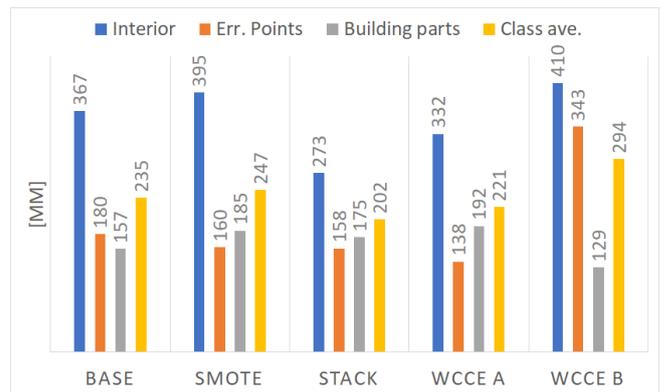


FIGURE 22. Geometric accuracy of the class combination 2-1 (subset A). The geometric accuracy is expressed by the SDFP points.

to 11% for these methods. In this context, the SA method is the only augmentation method with an improvement in both parameters, RP and PP (Table 7). The PP of the class Building Parts is very high with as values of 96% and 97% for all methods. However, the PP of the Interior is very low with a value of approximately 30% for all methods (Table 7). Points of the class Building Parts are sorted to a greater extent into the class Interior. This can also be seen in the SDFP points for the Interior, which are larger than 2000 mm (Fig. 24).

Subset B contains more different spaces with larger dimensions, so that larger SDFP points are also possible, as shown in Fig. 24. Furthermore, it can be observed that the SDFP points does not increase with larger rooms in subset B. Compared to subset A with erroneous points, this parameter even decreases.

TABLE 7. Semantic accuracy of the class combination 2-2 (subset B). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.46	0.00	0.00	0.46	0.46
Precision					
Interior	33%	33%	36%	32%	35%
Build. Parts	96%	96%	96%	97%	96%
Class average	65%	65%	66%	65%	66%
Recall					
Interior	63%	68%	66%	↑ 74%	60%
Build. Parts	91%	89%	89%	87%	93%
Class average	77%	78%	78%	81%	76%

The data augmentation methods do not lead to any increase in semantic and geometric accuracy for this class combination. It can be observed that the PP of the infre-

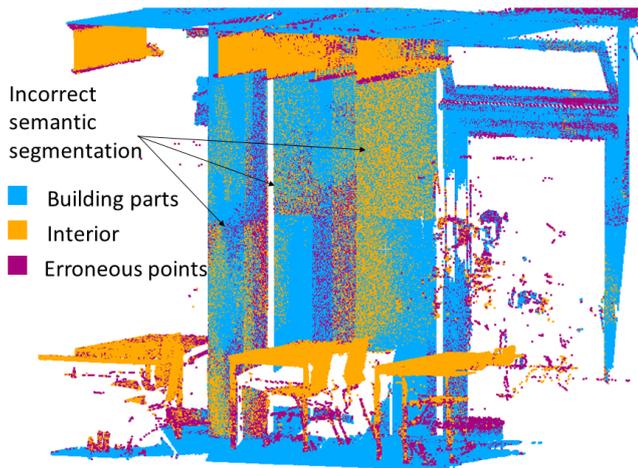


FIGURE 23. Example of major semantic segmentation errors in class combination 2-1. The parts of the wall (class Building Parts) become a segment of the class Interior.

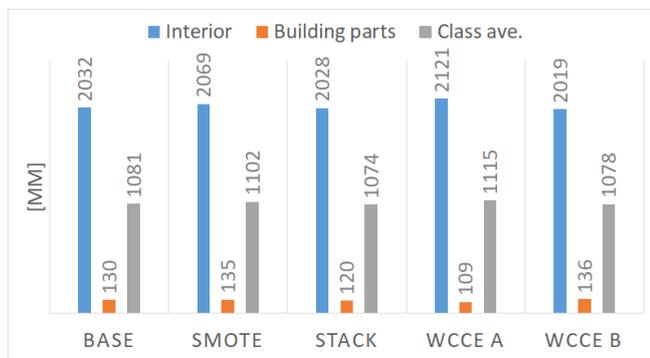


FIGURE 24. Geometric accuracy of the class combination 2-2 (subset B). The geometric accuracy is expressed by the SDFP points.

quent classes is not increased. The applied methods only increase the recognizability of the infrequent classes, but the discrimination is not increased. The reason for this is a lack of generalizability of the Base method for the used dataset. Rooms in the dataset differ strongly in terms of completeness, object surfaces, object geometries and sizes. An examination of the individual rooms shows that rooms with a 20 m x 20 m floor space, in which the scanner is positioned in the center, are best semantically segmented. In these rooms, there are usually only tables and chairs. Here, the RP and PP are higher than 88% for all methods. In rooms with rare objects, such as shelves, the semantic accuracy is usually less than 50%. The conditions in the different rooms influence segmentation quality strong.

The **combination 2-1** and **2-2** can not be compared, because they consist of different rooms. For a comparison the subset A without erroneous points is therefore used. The results for this subset are shown in Table 8. The comparison of these data with (Table 6) and without (Table 8) the class Erroneous Points shows that the absence of this class leads to an increase of up to 23% in the semantic accuracy of the object classes. An improvement through the data augmentation methods cannot be identified in the presented investigation.

TABLE 8. Semantic accuracy of the class combination 2-2 (subset A).

	Base	SMOTE	SA	WCCEa	WCCEb
Precision					
Interior	93%	93%	92%	92%	92%
Build. Parts	95%	94%	94%	95%	95%
Class average	94%	94%	93%	93%	94%
Recall					
Interior	90%	88%	88%	90%	90%
Build. Parts	96%	97%	96%	96%	96%
Class average	93%	92%	92%	93%	93%

C. CLASS COMBINATIONS 3-1, 3-2, 3-3 AND 3-4

The class Building Parts is further subdivided to distinguish individual building parts (application level). The choice of classes is based on those frequently used in as-built models or in BIM applications [97], such as defined in the IFC standard [98]. The subdivision is carried out for two levels in order to examine if combining the infrequent classes door and window leads to a better semantic segmentation.

In the combination 3-1 all building parts (floor, ceiling, window, door and wall), as well as the erroneous points are included. Combination 3-2 is identical to combination 3-1 without the class Erroneous Points. In combination 3-4 the infrequent classes Window and Door are combined as Opening (level 3), all other classes remain unchanged as in combination 3-1. The combination 3-3 is identical to combination 3-4 without the class Erroneous Points. All class combinations are shown in Table 1.

For **combination 3-1**, non of the methods meet the requirements (Table 9). The CE rate for the Base, WCCEa and WCCEb methods reaches a high value of 0.84. The SMOTE method shows the optimal class distribution and the SA method improves the rate to a moderate score of 0.52.

TABLE 9. Semantic accuracy of the class combination 3-1 (subset A). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.84	0.00	0.52	0.84	0.84
Precision					
Floor	99%	↓ 70%	99%	98%	99%
Ceiling	98%	↓ 44%	99%	99%	99%
Err. Points	75%	↓ 21%	75%	68%	↓ 63%
Window	39%	↓ 5%	42%	39%	↓ 29%
Door	53%	↓ 11%	58%	54%	52%
Wall	90%	83%	90%	88%	94%
Class average	76%	↓ 39%	77%	74%	73%
Recall					
Floor	99%	↓ 10%	99%	99%	99%
Ceiling	99%	90%	99%	98%	99%
Err. Points	72%	↓ 60%	76%	73%	72%
Window	68%	↓ 3%	58%	62%	77%
Door	26%	↓ 3%	20%	30%	24%
Wall	79%	↓ 24%	85%	77%	↓ 61%
Class average	77%	↓ 32%	73%	73%	72%

The SMOTE method is not suitable for class combination 3-1, because the semantic accuracy is reduced compared to the Base method. The average RP is 32% and the average PP is 39%. The Base method and all other methods have a RP of more than 58% for the frequent classes. For the infrequent

class Door (< 1% of the dataset) the RP varies between 20% to 30%. This class cannot be learned by the methods as shown in Table 10.

The PP of the classes Floor and Ceiling is 99% for the data augmentation methods. In contrast, the PP of the class Window is very low (< 45%) for all methods, because many points, especially of the class Wall and Door are assigned due to the large geometrical similarity of this class (Table 9). This low semantic accuracy correlates with a low geometric accuracy for this class. The SDFP points is larger than 6000 mm, so that this semantic class occupies nearly the whole room.

Since the classes Window and Door are not learned by the methods, they are combined in an intermediate step to the class Opening. The idea behind the summary is, that this class could be subdivided in the case of a good semantic segmentation in a following step, without negatively affecting the class Wall.

TABLE 10. Semantic accuracy of the class combination 3-4 (subset A). The geometric accuracy is expressed by the SDFP points. The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.72	0.00	0.26	0.72	0.72
Precision					
Floor	99%	99%	99%	98%	99%
Ceiling	98%	99%	99%	98%	97%
Err. Points	62%	71%	↓42%	67%	↑77%
Wall	88%	90%	91%	90%	89%
Opening	35%	↑51%	44%	30%	36%
Class average	76%	82%	75%	77%	80%
Recall					
Floor	95%	99%	99%	97%	99%
Ceiling	98%	96%	98%	95%	99%
Err. Points	71%	77%	↓60%	66%	71%
Wall	77%	↑90%	↑88%	73%	76%
Opening	54%	51%	↓41%	↑64%	63%
Class average	79%	83%	77%	79%	82%

For **combination 3-4** the Base, WCCEa and WCCEb methods meet the requirements with restrictions. The method SA does not meet the requirements and the SOMTE class meets the requirements. The CE rate for the Base, WCCEa and WCCEb methods is high with a value of 0.72. The SMOTE method shows the optimal class distribution and the SA method improves the CE rate to a sufficient score of 0.26. Due to the rough description of the distribution for these combination, it can be seen that an increase of the semantic accuracy is achieved (Table 10). The recognizability of the class Opening is low compared to the other classes. The PP of this class is for almost any method below 50%, and the SDFP points is higher than 3600 mm. Nevertheless, a good semantic segmentation can be performed with the SMOTE method. But it does not work for the combination 3-1.

For combinations 3-1 and 3-4, the TL phase leads to results comparable to the Base method.

For **combination 3-2**, no method meet the requirements. The CE rate for the Base, WCCEa and WCCEb methods is with 0.72 high. The SMOTE method shows the optimal

class distribution and the SA method improves the rate to a moderate score of 0.38 (Table 11).

TABLE 11. Semantic accuracy of the class combination 3-2 (subset B). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.72	0.00	0.38	0.72	0.72
Precision					
Floor	97%	95%	96%	97%	98%
Ceiling	98%	96%	95%	97%	97%
Window	20%	16%	18%	21%	20%
Door	33%	25%	↓21%	23%	28%
Wall	76%	↑86%	↑86%	↑87%	↑87%
Class average	65%	64%	63%	65%	58%
Recall					
Floor	99%	97%	↓77%	94%	↓77%
Ceiling	72%	↑89%	↑95%	↑91%	81%
Window	46%	↓35%	↑68%	43%	51%
Door	4%	↑46%	↑30%	↑42%	↑37%
Wall	59%	↓45%	↓39%	↓40%	↓42%
Class average	56%	62%	62%	62%	58%

For combination 3-2, the majority of the points of the classes Door and Window are not assigned to the correct classes. In addition, the geometrically similar class Wall is less recognized compared to combinations 3-1 and 3-4. The PP of Door and Window is low with a maximum of 33% over all methods (Table 11). The geometric accuracy of the two classes has a high SDFP points. For the class Window, the SDFP points is larger than 4300 mm and for the class Door it is larger than 3600 mm. Based on these evaluation parameters, it can be stated that the class distribution has no influence in this case. A semantic segmentation with the class combination 2-2 leads to a high semantic and geometric accuracy only for the classes Floor and Ceiling. Also, the combination of the classes Door and Window to Opening in an intermediate step is tested in combination 3-3, too.

For **combination 3-3**, the methods Base, SA and WCCEb meet the requirements. The SMOTE and the WCCEa methods meet the requirements with restrictions. The CE rate for the Base, WCCEa and WCCEb methods is with value of 0.40 moderate. The SMOTE method shows the optimal class distribution and the SA method improves the rate to an score of 0.08. The class distribution becomes favorable after the consolidation (Table 12).

The combination 3-3 leads to an increase in the semantic segmentation accuracy of all classes. The RP of the class Opening is higher than 50% for all methods. The class Wall, with which the class Opening is often confused, is correctly recognized only by SMOTE and WCCEa of the point majority. Based on the low PP of 29% to 32%, the confusion with the class Opening is confirmed (Table 12). Even with this combination, the neighboring classes Wall and Opening cannot be accurately separated. Larger variations for different rooms are observed here, but there are no rooms that can be segmented semantically very accurately. An influence of the class Erroneous Points is not observed.

A TL with the Base method results in an increase of 1% to 3% for RP and PP for all methods and classes.

TABLE 12. Semantic accuracy of the class combination 3-3 (subset B). The symbols \uparrow and \downarrow indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.40	0.00	0.08	0.40	0.40
Precision					
Floor	98%	95%	94%	97%	97%
Ceiling	96%	95%	94%	96%	95%
Wall	89%	81%	84%	86%	88%
Opening	29%	30%	31%	32%	31%
Class average	78%	75%	76%	78%	78%
Recall					
Floor	68%	\uparrow 99%	\uparrow 99%	\uparrow 92%	\uparrow 96%
Ceiling	83%	\downarrow 69%	\uparrow 95%	85%	79%
Wall	34%	\uparrow 51%	\uparrow 45%	\uparrow 52%	\uparrow 46%
Opening	83%	\downarrow 59%	\downarrow 65%	\downarrow 67%	73%
Class average	67%	69%	76%	74%	74%

D. SUMMERY AND OVERALL FINDINGS

The results show for the investigated settings, class definition and class combination, that two examined DHPs have only a minor influence on the semantic and geometric accuracy of semantic segmentation. The applied augmentation methods lead to an improved recognition of the infrequent classes. In the classification step, points are more often assigned to an infrequent class. This leads to a reduction in PP of infrequent classes. The SDFP points remains unchanged or even decreases due to the used augmentation methods. The PP of the segmentation improves stronger for frequent classes.

The number of classes itself has no influence on the semantic segmentation performance. Instead, the geometric similarity and the distance of the objects are important for distinguishing classes. The classes Floor and Ceiling can be well distinguished because of the large geometric distance (no shared boundary), whereas the classes Window and Wall are difficult to distinguish. When defining a class, the geometric distinguishability of the objects must be taken into account. This must be valid for the entire dataset, since rooms, for example, vary strong in size, shape and furnishing.

Using a class combination without erroneous points leads to an increase in PP and RP for the classes that already have a higher PP in a semantic segmentation with the class Erroneous Points. Classes that have a lower semantic PP in the semantic segmentation with erroneous points are recognized worse without this class and have a lower PP.

Applying an additional TL phase, where the previous result serves as a starting point for training with the Base method, does not lead to an increase in accuracy. For the SMOTE and SA methods, it result in less frequent detection of the infrequent classes and a similar performance as with the Base method.

VII. CONCLUSION AND OUTLOOK

The performance of DL-methods in semantic segmentation is influenced among other factors by HPs. In this work, the DHP, class combinations and methods to minimize the unbalanced classes have been studied. For the investigation, an AEE has been developed in which the established PointNet

architecture has been implemented.

The class combinations were organized in a hierarchic order, so that a semantic segmentation is performed only for a particular part of the point cloud, for combinations in level 3 and 4. Infrequent classes were combined and semantically segmented afterwards. This resulted in higher semantic and geometric accuracy for the class Building Parts and its frequent sub classes. The class Erroneous Points leads to a slightly higher semantic accuracy for infrequent classes.

The use of two data-based augmentation methods and two algorithm-based methods only achieved a small increase in semantic recognition. The applied methods usually increase the RP, so that the infrequent classes are recognized more often and the more frequent classes become more precise. This is advantageous for the combinations in level 1 and 2, because only the more frequent classes are needed for a building modeling.

The primary goal of this work is to increase RP and PP to over 50% for all classes using the augmentation methods. This goal was only achieved for the combination 3-4 with the SMOTE method. An increase in RP to a value higher than 50% is achieved with the SMOTE method additional four times, whereas the WCCEa method fulfills it for five of the seven combinations. This increase of the RP is achieved four times with the WCCEb method. The SA method results in an increase in RP and PP, but less than 50% in most cases. With the Base method, a RP of all classes higher than 50% was achieved twice. The primary goal was partly achieved.

In the course of the investigation, it was discovered that the geometric similarity of classes must be considered when forming the class combinations. Also, the choice of LNB has a large impact on the segmentation performance. Based on our observations, the choice of the local neighborhood and the differences between the individual rooms in the dataset are highly influential. The focus of further investigations should be on these DHPs. The influence of data augmentation methods is measurable, but currently of little relevance according to our sample BIM application. In terms of augmentation methods, we plan to examine the impact of US methods as well as a combination of US methods, OS methods and weighted loss functions.

ACKNOWLEDGEMENT

Many thanks to the annotators: Clemens Semmelroth, Stefanie Stand and Olga Konkova. Special thanks to Annette Scheider, Lena Barnefske and Christopher Klocke for proof-reading.

APPENDIX A CLASS DEFINITION

The class definition for the two upper levels (Fig. 14) are shown in Table 13. The class definitions for the super-class Building Parts are summarized in Table 14. This class definition is developed for a semantic segmentation as a basis for creating a BIM model of a public building.

TABLE 13. Semantic class definitions for the classes of two top levels.

Class name	Sub-classes	Description
Object	Building Parts, Interior	Points of the class Objects describe a true object. They describe a surface with a small variation of a few millimeters per surface (< 10 mm).
Erroneous Points		Points of the class Erroneous Points are individual points, that appear in obscured places, that represent tails on edges and (measurement) noise around smooth surfaces (> 10 mm).
Building Parts	Door, Ceiling, Floor, Wall, Window, Opening	Points of the class Building Parts include all points that belong to the building structure. Not including: switches, lamps or boards.
Interior		Interior objects are all objects that have been brought into the building or installed in the building after the shell has been completed. Examples are switches, vents, furniture, decoration, people or measuring equipment.

TABLE 14. Semantic class definition for classes of the super-class Building Parts.

Class name	Sub-classes	Description
Wall		The class Wall is the vertical shell of a room. It can be hidden by furnishing objects. The class Wall includes baseboards. Frames of windows and doors are the horizontal boundaries. In the vertical direction, the wall is delimited by intersections with the ceiling and the floor. Window frames are not part of the wall. Free-standing columns are part of the wall.
Floor		The class Floor is defined by the lowest points that span a horizontal plane. This plane can be hidden by furnishings. Its extension is bordered by the vertical walls. Points count as a part of a plane if they do not deviate from the plane by more than 5 mm.
Ceiling		The class Ceiling is defined by the top points that span a horizontal plane. This plane can be hidden by lamps or other interior objects. It is bounded by the wall in the vertical direction. Points count as a plane if they do not deviate from the plane by more than 5 mm.
Window		Points in the class Window describe the window frames. Points in the glass areas are considered to be disturbances (erroneous points). No distinction is made between windows that can be opened and those that cannot be opened. Window sills belong to the class Window.
Door		Points of the class Door can belong to the door leaf or the door frame. The viewing windows next to the door leaf belong to the class Door.
Opening	Door, Window	Combination of classes Window and Door.

REFERENCES

- [1] J. Zhao, X. Zhang, and Y. Wang, "Indoor 3d point clouds semantic segmentation bases on modified pointnet network," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIII-B2-2020, pp. 369–373, aug 2020.
- [2] M. Soilán, R. Lindenbergh, B. Riveiro, and A. Sánchez-Rodríguez, "Pointnet for the automatic classification of aerial point clouds," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2/W5, pp. 445–452, may 2019.
- [3] S. D. Geyter, J. Vermandere, H. D. Winter, M. Bassier, and M. Vergauwen, "Point cloud validation: On the impact of laser scanning technologies on the semantic segmentation for BIM modeling and evaluation," *Remote Sensing*, vol. 14, no. 3, p. 582, jan 2022.
- [4] F. Noichl, A. Braun, and A. Borrmann, "'bim-to-scan' for scan-to-bim: Generating realistic synthetic ground truth point clouds based on industrial 3d models," in *Proceedings of the 2021 European Conference on Computing in Construction*. University College Dublin, jul 2021, pp. 164 – 172.
- [5] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang, "Hierarchical aggregation for 3d instance segmentation," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2021, pp. 15 447–15 456.
- [6] F. Poux and R. Billen, "Voxel-based 3d point cloud semantic segmentation: Unsupervised geometric and relationship featuring vs deep learning methods," *ISPRS International Journal of Geo-Information*, vol. 8, no. 5, p. 213, may 2019.
- [7] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3d: A review of point cloud semantic segmentation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 8, no. 4, pp. 38–59, dec 2020.
- [8] C. Morbidoni, R. Pierdicca, R. Quattrini, and E. Frontoni, "Graph cnn with radius distance for semantic segmentation of historical buildings tls point clouds," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLIV-4/W1-2020, pp. 95–102, sep 2020.
- [9] L. Winiwarter and G. Mandlbürger, "Classification of 3d point clouds using deep neural networks," in *Dreiländertagung der DGPF, der OVG und der SGPF in Wien, Österreich – Publikationen der DGPF, Band 28, 2019, 2019*, pp. 663–674.
- [10] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao, "Are we hungry for 3d LiDAR data for semantic segmentation? a survey of datasets and methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6063–6081, jul 2022.
- [11] H. Riemenschneider, A. Bodis-Szomorú, J. Weissenberg, and L. V. Gool, "Learning where to classify in multi-view semantic segmentation," in *2014 European Conference on Computer Vision ECCV*. Springer International Publishing, 2014, pp. 516–532.
- [12] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *International Conference on Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ*. IEEE, 2015, pp. 922–928.
- [13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017*, pp. 77–85.
- [14] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-net: Efficient semantic segmentation of large-scale point clouds," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020, pp. 11 105–11 114.
- [15] T. Bender, M. Härtig, E. Jaspers, M. Krämer, M. May, M. Schlundt, and N. Turianskyj, "Building information modeling," in *CAFM-Handbuch*. Springer Fachmedien Wiesbaden, 2018, ch. 11, pp. 295–324.
- [16] DIN18710, "Engineering survey," Sep. 2010.
- [17] M.-O. Löwner and G. Gröger, "Das neue lod konzept für citygml 3.0," in *13. GeoForum MV 2017, Rostock-Warnemünde*, vol. 13, 04 2017, pp. 23–30.
- [18] buildingSMART, "Industry foundation classes 4.0.2.1." Online available: <https://standards.buildingsmart.org> visited 24. June 2021, 2021.
- [19] E. S. Malinverni, R. Pierdicca, M. Paolanti, M. Martini, C. Morbidoni, F. Matrone, and A. Lingua, "Deep learning for semantic segmentation of 3d point cloud," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W15, pp. 735–742, aug 2019.
- [20] D. Passos and P. Mishra, "A tutorial on automatic hyper-parameter tuning of deep spectral modelling for regression and classification tasks," *Chemometrics and Intelligent Laboratory Systems*, vol. 223, p. 104520, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169743922000314>
- [21] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," *arXiv preprint arXiv:1502.02127*, 2015.
- [22] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay," *preprint arXiv*, 2018.
- [23] C. Cooney, A. Korik, R. Folli, and D. Coyle, "Evaluation of hyperparameter optimization in machine and deep learning methods for decoding imagined speech EEG," *Sensors*, vol. 20, no. 16, p. 4629, aug 2020.
- [24] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," *Hyper-Parameter Optimization: A Review of Algorithms and Applications*, 2020.
- [25] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine Learning*. Springer International Publishing, 2019, pp. 3–33.
- [26] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *IEEE/CVF International Conf. on Computer Vision (ICCV)*. IEEE, oct 2019, pp. 9296–9306.
- [27] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu, "Shape2motion: Joint analysis of motion parts and attributes from 3d shapes," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2019, pp. 8868–8876.
- [28] W. Zimmer, A. Rangesh, and M. Trivedi, "3d bat: A semi-automatic, web-based 3d annotation toolbox for full-surround, multi-modal data streams," in *IEEE Intelligent Vehicles Symposium (IV)*. IEEE, jun 2019, pp. 1816–1821.
- [29] M. Weinmann, B. Jutzi, C. Mallet, and M. Weinmann, "Geometric features and their relevance for 3d point cloud classification," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1/W1, pp. 157–164, may 2017.
- [30] M. Negassi, D. Wagner, and A. Reiterer, "Smart(sampling)augment: Optimal and efficient data augmentation for semantic segmentation," *Algorithms*, vol. 15, no. 5, p. 165, may 2022.
- [31] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, mar 2019.
- [32] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 286–304, jul 2015.
- [33] M. Weinmann, B. Jutzi, and C. Mallet, "Semantic 3d scene interpretation: A framework combining optimal neighborhood size selection with relevant features," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. II-3, pp. 181–188, aug 2014.
- [34] T. Hackel, "Large-scale machine learning for point cloud processing," Ph.D. dissertation, ETH Zürich, 2018.
- [35] N. Shukla, *Machine Learning with Tensorflow*. manning pubn, 2018.
- [36] E. Camuffo, D. Mari, and S. Milani, "Recent advancements in learning algorithms for point clouds: An updated overview," *Sensors*, vol. 22, no. 4, p. 1357, feb 2022.
- [37] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3d.net: A new large-scale point cloud classification benchmark," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1-W1, pp. 91–98, may 2017.
- [38] S. Wirges, T. Fischer, C. Stiller, and J. B. Frias, "Object detection and classification in occupancy grid maps using deep convolutional networks," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3530–3535.
- [39] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner, "Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans," in *CVPR*, vol. 1. IEEE, jun 2018, p. 2.
- [40] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652 – 7660.
- [41] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors*, vol. 19, no. 19, p. 4188, sep 2019.
- [42] E. Barnefske and S. Harald, "Automatisch semantisch-segmentierte punktwolken – möglichkeiten und herausforderungen," in *DVW-Seminar MST 2022 – von (A)nwendungen bis (Z)ukunftstechnologien*, vol. 103. Wißner-Verlag, 2022, pp. 173–184.
- [43] D. Koguciuk, Łukasz Chechliński, and T. El-Gaaly, "3d object recognition with ensemble learning - a study of point cloud-based deep learning models," *preprint arXiv*, 2019.

- [44] S. A. Bello, S. Yu, and C. Wang, "Review: deep learning on 3d point clouds," *remote sensing*, vol. 12, no. 11, p. 1729, may 2020.
- [45] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108. [Online]. Available: <https://arxiv.org/pdf/1706.02413.pdf>
- [46] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3d semantic segmentation of point clouds," in *2017 IEEE International Conference on Computer Vision Workshops (ICCV)*. IEEE, Oct. 2017, pp. 716–724.
- [47] H.-I. Lin and M. C. Nguyen, "Boosting minority class prediction on imbalanced point cloud data," *Applied Sciences*, vol. 10, no. 3, p. 973, feb 2020.
- [48] Z. Jiang, T. Pan, C. Zhang, and J. Yang, "A new oversampling method based on the classification contribution degree," *Symmetry*, vol. 13, no. 2, p. 194, jan 2021.
- [49] J. V. Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proceedings of the 24th international conference on Machine learning - ICML '07*. ACM Press, 2007, p. 935–942.
- [50] J. Zhang and I. Mani, "knn approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of workshop on learning from imbalanced datasets*, vol. 126. ICML, 2003, pp. 1–7.
- [51] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proceedings of the 14th International Conference on Machine Learning*, 1997, pp. 179–186.
- [52] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 806–814.
- [53] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 40–49, jun 2004.
- [54] P. Hensman and D. Masko, "The impact of imbalanced training data for convolutional neural networks," 2015.
- [55] D. Griffiths and J. Boehm, "Weighted point cloud augmentation for neural network training data class-imbalance," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, pp. 981–987, jun 2019.
- [56] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *journal of artificial intelligence research*, vol. 16, 2002.
- [57] H. Lee, M. Park, and J. Kim, "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2016, pp. 3713–3717.
- [58] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, jul 2016, pp. 4368–4374.
- [59] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017, pp. 2999–3007.
- [60] H. Wang, Z. Cui, Y. Chen, M. Avidan, A. B. Abdallah, and A. Kroner, "Predicting hospital readmission via cost-sensitive deep learning," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 6, pp. 1968–1978, nov 2018.
- [61] C. Zhang, K. C. Tan, and R. Ren, "Training cost-sensitive deep belief networks on imbalance data problems," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, jul 2016, pp. 4362–4367.
- [62] Y. Zhang, L. Shuai, Y. Ren, and H. Chen, "Image classification with category centers in class imbalance situation," in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, may 2018, pp. 359–363.
- [63] S. H. Khan, M. Hayat, M. Bennamoun, F. Sohel, and R. Togneri, "Cost sensitive learning of deep feature representations from imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 8, pp. 3573–3587, aug 2018.
- [64] W. Ding, D.-Y. Huang, Z. Chen, X. Yu, and W. Lin, "Facial action recognition using very deep networks for highly imbalanced class distribution," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, dec 2017, pp. 1368–1372.
- [65] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, oct 2018.
- [66] J. Morel, A. Bac, and T. Kanai, "Segmentation of unbalanced and inhomogeneous point clouds and its application to 3d scanned trees," *The Visual Computer*, vol. 36, no. 10-12, pp. 2419–2431, sep 2020.
- [67] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 5375–5384.
- [68] S. Ando and C. Y. Huang, "Deep over-sampling framework for classifying imbalanced data," in *Machine Learning and Knowledge Discovery in Databases*. Springer International Publishing, 2017, pp. 770–785.
- [69] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 6, pp. 1367–1381, jun 2019.
- [70] T.-Y. Liu, "EasyEnsemble and feature selection for imbalance data sets," in *2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing*. IEEE, 2009, pp. 517–520.
- [71] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTE-Boost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003*. Springer Berlin Heidelberg, 2003, pp. 107–119.
- [72] O. Hassaan, A. Shamail, Z. Butt, and M. Taj, "Point cloud segmentation using hierarchical tree for architectural models," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2019, pp. 1582–1586.
- [73] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "SceneNN: A scene meshes dataset with aNnotations," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, oct 2016, pp. 92–101.
- [74] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia, "Hierarchical point-edge interaction network for point cloud semantic segmentation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2019, pp. 10 432–10 440.
- [75] Y. Li and G. Baciu, "HSGAN: Hierarchical graph learning for point cloud generation," *IEEE Transactions on Image Processing*, vol. 30, pp. 4540–4554, 2021.
- [76] 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, p. 102626, dec 2021.
- [77] T. H. Kolbe, T. Kutzner, C. S. Smyth, C. Nagel, C. Roensdorf, and C. Heazel, "Ogc city geophymarkup language(citygml) part 1: conceptual model standard," pen Geospatial Consortium, Tech. Rep., 2021.
- [78] Y. Verdier, F. Lafarge, and P. Alliez, "LOD generation for urban scenes," *ACM Transactions on Graphics*, vol. 34, no. 3, pp. 1–14, may 2015.
- [79] L. Tang, L. Li, S. Ying, and Y. Lei, "A full level-of-detail specification for 3d building models combining indoor and outdoor scenes," *ISPRS International Journal of Geo-Information*, vol. 7, no. 11, p. 419, oct 2018.
- [80] H. Ledoux, K. A. Otori, K. Kumar, B. Dukai, A. Labetski, and S. Vitalis, "Cityjson: A compact and easy-to-use encoding of the citygml data model," *Open Geospatial Data, Software and Standards*, vol. 4, no. 1, pp. 127–140, jun 2019.
- [81] X. Li, C. Li, Z. Tong, A. Lim, J. Yuan, Y. Wu, J. Tang, and R. Huang, "Campus3d: A photogrammetry point cloud benchmark for hierarchical understanding of outdoor scene," in *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, oct 2020, p. 238–246.
- [82] V. Stojanovic, H. Shoushtari, C. Askar, A. Scheider, C. Schuldt, N. Hellweg, and H. Sternberg, "A conceptual digital twin for 5g indoor navigation," in *The Eleventh International Conference on Mobile Services, Resources, and Users - MOBILITY 2021*, 04 2021.
- [83] Zoller+Fröhlich-GmbH, "Reaching new levels, z+f imager5016, user manual, v2.1," 2019.
- [84] CloudCompare, "3d point cloud and mesh processing software open-source project," Available: <http://www.cloudcompare.org/>. Accessed: Jun. 24, 2021, 2021, version 2.12.
- [85] Autodesk-Recap, "Youtube channel," Available: <http://https://www.youtube.com/user/autodeskreap/>. Accessed: Jun. 24, 2022, 2022.
- [86] E. Barnefske and H. Sternberg, "Evaluating the quality of semantic segmented 3d point clouds," *Remote Sensing*, vol. 14, no. 3, p. 446, jan 2022.
- [87] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3d point clouds: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, dec 2021.

- [88] S. Rakshit and S. Paul, "Point cloud segmentation with pointnet," GitHub, Oct. 2020, programm Code. [Online]. Available: <https://github.com/soumik12345/point-cloud-segmentation>
- [89] Z. Li, K. Kamnitsas, and B. Glocker, "Analyzing overfitting under class imbalance in neural networks for image segmentation," *IEEE Transactions on Medical Imaging*, vol. 40, no. 3, pp. 1065–1077, mar 2021.
- [90] M. Abdou, M. Elkhateeb, I. Sobh, and A. Elsallab, "End-to-end 3d pointcloud semantic segmentation for autonomous driving," 2019.
- [91] M. Barel, "Multi-class weighted loss for semantic image segmentation in keras/tensorflow," Available: <https://stackoverflow.com/questions/59520807/multi-class-weighted-loss-for-semantic-image-segmentation-in-keras-tensorflow>. Accessed: Sep. 23, 2022, Dec. 2019.
- [92] E. Grilli, D. Dinunno, G. Petrucci, and F. Remondino, "From 2d to 3d supervised segmentation and classification for cultural heritage applications," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2, pp. 399–406, may 2018.
- [93] S. Teruggi, E. Grilli, M. Russo, F. Fassi, and F. Remondino, "A hierarchical machine learning approach for multi-level and multi-resolution 3d point cloud classification," *Remote Sensing*, vol. 12, no. 16, p. 2598, aug 2020.
- [94] E. Barnefske and H. Sternberg, "Klassifizierung von fehlerhaft gemessenen punkten in 3d-punktwolken mit convnet," in *Ingenieurvermessung 20. Beiträge zum 19. Internationalen Ingenieurvermessungskurs München, 2020*, ser. 19, T. Wunderlich, Ed., no. 19. Herbert Wichmann Verlag, Mar. 2020, pp. 127–139.
- [95] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=ryQu7f-RZ>
- [96] C. Schuldt, H. Shoushtari, N. Hellweg, and H. Sternberg, "L5in: Overview of an indoor navigation pilot project," *Remote Sensing*, vol. 13, no. 4, p. 624, 2021.
- [97] J. Blankenbach, *Ingenieurgeodäsie*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, ch. Bauaufnahme, Gebäudeerfassung und BIM, pp. 23–53.
- [98] BIM-Forum, "Level of development specification part1 & commentary," Available: <https://bimforum.org/lod/>. Accessed: Dec. 8, 2020, 2020.



HARALD STERNBERG studied Surveying at the University of the Federal Armed Forces Germany, Munich and graduated in 1986. He worked in the administrative chain as an artillery officer and as a research assistant at the University of the Bundeswehr Germany, Munich. There he was graduated as Dr.-Ing. in 1999. In 2001, he became Professor of Engineering Geodesy at the Hamburg University of Applied Sciences and from 2009 to 2017 he was Professor of Engineering Geodesy and Geodetic Metrology at the HafenCity University Hamburg. At HafenCity University he was also Vice-President for Studies and Teaching from 2009 to 2022. In 2017 he took over the professorship for Hydrography and Geodesy. His research areas include mobile mapping systems on different carriers (cars, ships and indoor cars), the use of low-cost sensors for positioning, indoor positioning, including with 5G, monitoring of structures, autonomous underwater vehicles, automatic analysis of underwater images, interpretation of backscatter data and analysis of mass data using artificial intelligence.

...



EIKE BARNEFSKE studied of Geomatics in the Bachelor and Master program of the HafenCity University Hamburg. He revised his M.Sc. degree in 2016. From 2016 to 2017, he worked as a research assistant for Engineering Geodesy and Geodetic Metrology. Since 2017 he is research assistant and PhD-Student for Hydrography and Geodesy. His research interests are the analysis of mass data, such as laser scanning data, and the development of multi-sensor-systems.

B Non-peer-reviewed publication

B.1 PCCT: A Point Cloud Classification Tool To Create 3D Training Data To Adjust And Develop 3D ConvNet

Reference:

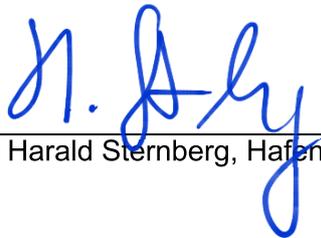
E. Barnefske & H. Sternberg (2019): PCCT: A Point Cloud Classification Tool to Create 3D Training Data to Adjust and Develop 3D ConvNet, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-2/W16, 35–40, <https://doi.org/10.5194/isprs-archives-XLII-2-W16-35-2019>.

Contribution of Co-Authors:

Table 9: Contribution to Paper No. 0

Involved in	Estimated contribution
Ideas and conceptual design	90%
Computation and results	100%
Analysis and interpretation	95%
Manuscript, figures and tables	100%
Total:	96%

I hereby confirm the correctness of the declaration of the contribution of Eike Barnefske for Paper 0 in Table 9:



Prof. Dr.-Ing. Harald Sternberg, HafenCity Universität Hamburg

PCCT: A POINT CLOUD CLASSIFICATION TOOL TO CREATE 3D TRAINING DATA TO ADJUST AND DEVELOP 3D CONVNET

E. Barnefske, H. Sternberg

Geodesy and Geoinformatic, HafenCity University Hamburg, Germany- (eike.barnefske, harald.sternberg)@hcu-hamburg.de

ICWG II/III: Pattern Analysis in Remote Sensing

KEY WORDS: ConvNet, semantic labeling, training data, TLS, deep learning

ABSTRACT:

Point clouds give a very detailed and sometimes very accurate representation of the geometry of captured objects. In surveying, point clouds captured with laser scanners or camera systems are an intermediate result that must be processed further. Often the point cloud has to be divided into regions of similar types (object classes) for the next process steps. These classifications are very time-consuming and cost-intensive compared to acquisition. In order to automate this process step, conventional neural networks (ConvNet), which take over the classification task, are investigated in detail. In addition to the network architecture, the classification performance of a ConvNet depends on the training data with which the task is learned. This paper presents and evaluates the point cloud classification tool (PCCT) developed at HCU Hamburg. With the PCCT, large point cloud collections can be semi-automatically classified. Furthermore, the influence of erroneous points in three-dimensional point clouds is investigated. The network architecture PointNet is used for this investigation.

1. INTRODUCTION

Complex and unsorted point clouds are often used to visualize the results of a survey recorded by laser scanners or camera systems. These point clouds give a very detailed and sometimes highly accurate representation of the geometry of the captured objects. A human observer can recognize the captured objects in the point cloud and separate them from each other. In addition, incorrect measurements, such as mixed pixels and multipath effects, which result from the acquisition technique can be detected and eliminated. Solving this complex and time-consuming task of semantic segmentation and classification through an automated process is a key challenge in processing large point clouds into detailed models. A promising approach to automate this task is the usage of convolutional neural networks (ConvNets). Simply expressed, specific features in the point cloud are identified by ConvNets and according to meaning of the features each point of the point cloud is assigned to a predefined class. A ConvNet can be considered as a very large number of simple functions for extracting the features that are chained to each other. The results of the functions are weighted to improve the classification so that the error between true class and the class predicted by the ConvNet is minimal over all points (network learning). This learning requires a large amount of classified point clouds to optimize the network weights. For point clouds resulting from surveys, the challenge is also to distinguish measurement errors from true points. Therefore a high quality of the classification is critical.

The preview version of Point Cloud Classification Tool (PCCT) presented in Barnefske & Sternberg (2019) will be enhanced in this paper. This tool is used to generate efficient and reliable test and training data for point clouds classification applications. The basic idea of PCCT is to project colored point clouds in the two dimensional space, generate segments out of the points that describe different objects, classify the segments and back project the information on the three dimensional point cloud (Fig. 1). The development and the evaluation of PCCT was proceeded by our HafenCity point set. HafenCity point set is

a set of indoor and outdoor point clouds that manually classified and captured by a terrestrial laser scanner.

In the second part of this paper we evaluate the ConvNet architecture PointNet (Qi et al., 2017a) with HafenCity (HC) point set. PointNet is a ConvNet for semantic classification of indoor areas. The main goal of the investigation is the determination of the influence that erroneous points in points cloud have for the classification performance.

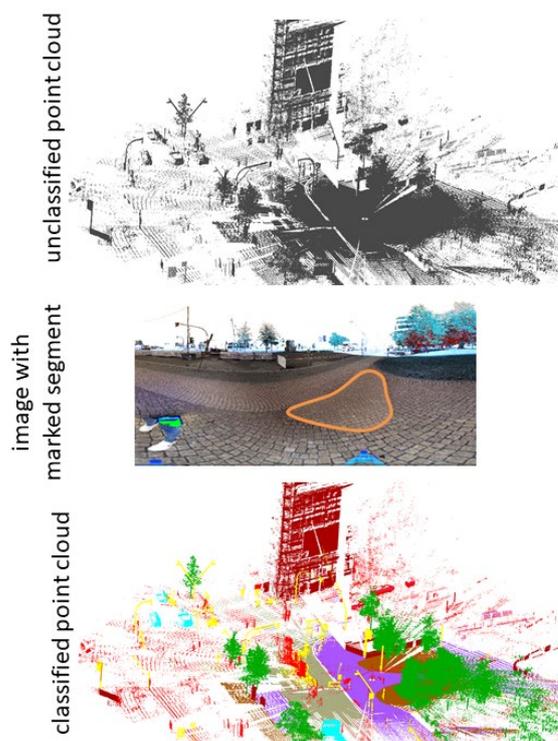


Figure 1. Basic idea of PCCT.

2. STATE OF THE ART

2.1 CNN for 3D-Point Classification

The results of a survey usually have to be processed further to make them a basis for decisions and planning. Today, numerous surveys are carried out with measurement systems that represent the geometry of objects and their surroundings as a digital point cloud. Point clouds usually represent several objects and their surroundings due to the recording conditions. Therefore, single objects have to be separated and assigned to a sense class. This step is often called segmentation and classification. In classical procedures, such as edge-based, graph-based or hierarchical segmentation, segmentation can be clearly differentiated from the classification task that assigns a class to the segment. Classifications are traditionally performed by the human operator. Grilli et al. (2019) give a brief overview of segmentation methods without the use of artificial intelligence (AI). In the following section, two of these methods will be explained in more detail relating to the PCCT.

An efficient processing of highly inhomogeneous mass data, such as three-dimensional point clouds, is with traditional processing methods cost-intensive and time-consuming. Applying AI methods, such as artificial neural networks, allows to solve predictions and classifications more efficiently and sometimes more accurately than by a human operator. For the classification of point clouds, the aim is to combine points with the same characteristics into a sense class. For this purpose, characteristics in the point clouds must be determined and due to their similarity, they have to be summarized/classified in classes that are predefined (supervised learning) and freely formed (unsupervised learning). With large data sets, which are two- or multidimensional, convolutional neural networks (ConvNet) are successfully used for feature extraction (Fig. 2).

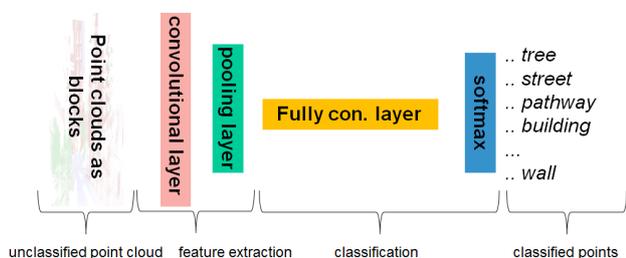


Figure 2. Structure of ConvNet for semantic point cloud classification.

Using ConvNets, Girsick (2015) and Redmon et al. (2016) have performed precise classifications of objects in images, which have ensured that ConvNets are applied to multidimensional data sets. Some approaches project three-dimensional point clouds into two-dimensional space and apply the established 2D ConvNet architectures. These methods are disadvantageous for applications where the entire 3D point cloud needs remain geometrically unchanged. In addition, it is often not possible to separate objects lying behind each other and a generalization of the data is unavoidable.

Besides pixels, the data structure of voxels is excellent for applying ConvNets to them. Voxels represent an even grid, similar to pixels in a digital photo. Adjacent points that fall into a voxel can be combined and the entire object space is represented by an even voxel grid. Maturana & Scherer, (2015) use this voxel structure and apply ConvNets similarly as with images. In combination with occupancy grids, this

structure becomes more efficient (Wirges et. al 2018). Hackel et al. (2017) combine voxel grids of different sizes to classify terrestrial laser scan data with different densities.

In addition to classification methods with ConvNets, that use a regular data structure, there are approaches to structure the point cloud through the net itself. In this case, the unstructured and differently dense point cloud must not be transformed into an auxiliary structure. This approach was used for the first time in the network architecture named PointNet (Qi et al., 2017a). In PointNet, the point clouds are given blockwise into ConvNet. In a block, the points are freely distributed and features are extracted based on their geometry. These characteristics are used for classification. Thereby, local and global characteristics are determined and combined within a block. From the mix of local and global characteristics, the classification is then carried out in the classification step. The biggest drawback of this network is that the information can only be used in one block. Engelmann et al. (2017) counter this with a scalable block and the simultaneous processing of several blocks (changes to the data inputs). Furthermore, information from previous data passes are also used as input information. Another enhancement is PoinNet++ (Qi et al., 2017b) in which PointNet is extended by segmentation and grouping layers.

2.2 Training Data for CNN Classification

Most of the current developments are based on synthetic data, because synthetic 3D data can be generated faster from models and this data automatically has some reliable ground truth data for training and evaluation. With the focus on (real) 3D point clouds from a LIDAR scanner, there are only a handful of data sets available, which have been consisted mainly of scans from low-cost laser scanners. The *KITTI* data set (Geiger et al., 2012) is one of the most popular data set consisting of synthetic and measured data. This data can be used for applications in surveying, like mobile mapping. A similar data set is captured by the Velodyne HDL-64E LIDAR-Scanner and can be found in Gehring et al. (2017). The *Semantic3D.Net* data set (Hackel et al., 2017) consists of 31 high-quality and classified terrestrial panoramic laser scans. To the best of our knowledge the data of this set uses similar raw data as our tool.

2.3 HafenCity Point Cloud Set

The investigation and the development of the PCCT were carried out with a point cloud data set consisting of 9 point clouds for indoor and 9 point clouds for outdoor scenes. The point clouds were captured with the laser scanner Zoller + Fröhlich 5010 with a resolution setting of 6 mm at 10 m. In the post-processing the point clouds were colored by panorama images, which were created from the same position. The point clouds were not filtered and neighboring point clouds are connected by target signs, which were installed in the object space during the capture. The point cloud set consists of about 117 million measured points and is almost completely manually classified according to the criteria object classes or error classes.

3. PCCT

The quality of a classification, by a human or a machine depends primarily on the data used for classification. The more heterogeneous data sets are used for the classification application, the more reliable the classification result will be in general. In addition to the number of data sets (here number of

points), the number of available features of a data set is very important for a classification. With current laser scanners with one or more cameras integrated, color information for each point is available in addition to the geometric information and intensities. These color information are especially necessary for the manual classification of point clouds when detailed objects are segmented and assigned to a class in a further step. Automated segmentation methods for two-dimensional images have reached a high degree of sophistication, so that segments of objects in images can be generated reliably and accurately. The classification of heterogeneous objects in case of a small amount of data, presents a great challenge for automation. Especially, in the case when several segments describe one object. However, this task can usually be reliably performed by a human. Based on this knowledge, the PCCT is used to process point clouds for the training purpose of ConvNets.

3.1 Concept and Method of Operation

The PCCT is a tool for semiautomatic classification of point clouds. The motivation for this tool is that the segmentation of objects in a three-dimensional point cloud is very time-consuming and depends on the skills/ interpretation of the user. With the PCCT the segmentation step is automated and the classification is efficiently possible by any large number of non-trained users. Colored point clouds are better suited for the application in PCCT, because on one hand the segmentation is based on color values and on the other hand the human classifiers can better recognize objects and assign them to a class due to the additional colored information. The PCCT can be divided into three process steps: point cloud transformed to image and segmentation, classification in a web application and applying the classification to the point clouds (Fig. 3).

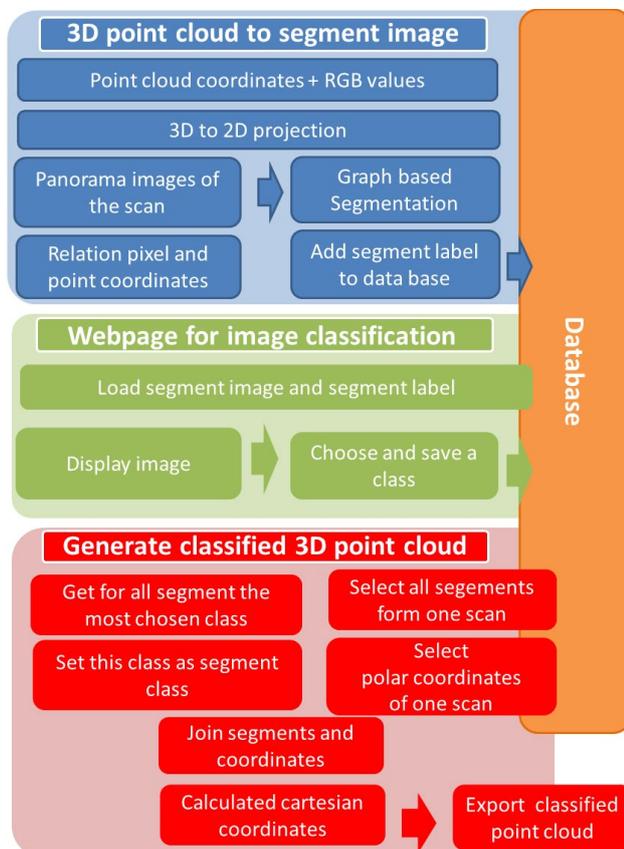


Figure 3. Workflow and main modules of PCCT.

In the first process step, the point cloud of a terrestrial laser scan is projected into the two-dimensional space so that two panoramic images are calculated. It has been shown, that the division of the scan into two halves allows a more efficient calculation of the panoramic images as well as a more accurate generation of segments. Different resolutions for the projection were investigated. The resolution is crucial for the level of detail of the segments. Image resolutions of 600 x 500 px or 800 x 700 px enable the segmentation of small elements, such as table legs, and avoid a too detailed segmentation of large surfaces, such as floors with lower inhomogeneity. Furthermore, the influence of distortion in the outer areas is kept to a minimum. With colored scans, the color value of the points is assigned to the pixel in which the points fall. If no color information is available, depth or intensity images can be calculated. However, these are more difficult for the human classifier to interpret, so that only colored scans are used in the investigations.

The segmentation was examined using two algorithms. In the first PCCT version, an edge detection method was used in combination with the Rosebrock (2015b) watershed algorithm. With this method, the edge image of Rosebrock (2015a) is calculated from the panorama using the canny algorithm. The edge image is placed as a mask on the panorama so that segments are generated. Each segment is provided with a segment label. The segment label is assigned to each point via the relationship between pixels and points. In the further process, the segment label is replaced by a class label. A detailed description can be found in Barnefske & Sternberg (2019). This segmentation method does not allow a clear separation of the segments, especially for small objects in the image, so that an alternative graph-based method is implemented in the PCCT for segmentation

In the current version of PCCT, segmentation is performed with a graph-based algorithm according to Felzenszwalb & Huttenlocher (2004). The algorithm spans a graph over the entire image and weights the edges of the graph. The edge weights are calculated through the distance between two pixels in the feature space (e.g. color value difference). The weights are sorted and pixels with similar weights are combined to one segment. A threshold value is calculated on the basis of the color values shown in the image. The minimum segment size parameter is used to avoid very small segments. With this simple algorithm it is possible to increase the resolution for the segments and simultaneously minimize the number of segments per panorama image. As in the first PCCT version, each segment is labelled and assigned to the points (Fig. 3, sect. 3).

The classification of the segments is based on the panorama images in which the segment to be classified is marked. All images are stored in a database together with the segment label. The panorama images are loaded via a web application and the marked segments can be assigned to one of 18 classes. Each classification for any segment label is stored in a results database (Fig. 3 sect. 2). In the web application, the images are displayed in a random order so all panorama images are evenly classified. A classification of subsets and the simultaneous classification by different users are possible.

After all segments of a point cloud collection are classified, the points will be classified via the segment label. The segment label is stored in the database in which the point cloud is stored

and in the results database as well. The segment label is used to assign each point to a class. If several classifications are available for one segment label, then the class that is most available for this segment is used. This can occur when users make a fatal error (selecting the wrong class) or interpret the segment differently. For an efficient processing the point clouds are exported as polar and cartesian coordinates with a class label. For the application on a ConvNet it has been shown that the output by classes in individual files is feeding, so this option has been enhanced (Fig. 3, sect. 3).

3.2 Investigation and evaluation

The classification of three-dimensional point clouds can be carried out by untrained users using the PCCT, because segmentation takes place automatically. The classification itself is done by classifying an area highlighted in an image via a dropdown menu in which predefined classes are listed. The PCCT is deliberately designed this way, so that no complex decisions are made by the classifier to avoid errors and to process large data sets efficiently. Large data sets of up to 10,000 segments (about 50 million points) can be classified in a few hours by several users simultaneously. This multi-user capability is designed to reduce the classification time for the individual user as well as to verify the classifications.

Colored and partially uncolored point clouds can be processed with the PCCT regardless of the recording sensor. The number of data classified with PCCT is easily scalable so that new data sets can be added or processed data sets can be removed from the database. In terms of usability, the PCCT is an essential and efficient component in the development process of data-based classification systems for three-dimensional point clouds.

The characteristics semantic correctness and accuracy of the PCCT are examined by means of a manually classified point cloud. For this verification the parameter precision for the characteristic accuracy and the parameter recall for the characteristic accuracy are determined. The precision (eq.1) refers to the set of points assigned to a class by the PCCT and represents the relationship between correctly classified points (TP) and incorrectly classified points (FP).

$$\text{precision} = \frac{TP}{TP + FP} \quad (1)$$

The parameter recall (eq. 2) refers to the number of points of one class in the reference data set and represents the ratio of correctly classified points (TP) and those not assigned to this reference class (FN).

$$\text{recall} = \frac{TP}{TP + FN} \quad (2)$$

As a benchmark for quality of classifications the intersection over union (IoU) is commonly used parameter (eq. 3). This describes the correctness (recall) and precision

$$\text{IoU} = \frac{TP}{TP + FN + FP} \quad (3)$$

The correctness and precision of the PCCT for multiple point clouds with about 16 million points is shown in Table 1. The scans used are outside scans divided into eight target classes of the reference data set. To all seven object classes points were assigned by the PCCT. No points were assigned to the class of

error points, because through the projection of the point cloud into the two-dimensional space, most of the error points were included in an object class lying in front of or behind. Erroneous points do not span large segments in the images. The lack of depth differentiability also leads to occasional errors in the classification of object classes which are represented by a low recall value (max. = 1). The boundary between two objects can only be dated roughly by the segmentation algorithm. The percentage of points that were precisely classified (max. = 1) is higher for large and plane object classes than for small object classes with more heterogeneous geometry.

class	precision	recall	IoU
building	0,85	0,65	0,59
car	0,90	0,60	0,56
floor veg	0,63	0,41	0,33
pathway	0,86	0,79	0,70
street	0,69	0,57	0,45
tree	0,88	0,39	0,37
sign	0,67	0,05	0,05
erroneous points	0,00	0,00	0,00
all classes	0,82	0,62	0,54

Table 1: Investigate the performance of the PCCT with a reference point cloud from the HC point set. Using the parameters of the precision, recall and intersection of union (IoU).

All scans are almost entirely classified, based on the number of points. Near areas showed a higher density due to the recording constellation. The parameter completeness leads to a misinterpretation, as shown in Figure 4. In the point cloud illustrated here, 93% of all points are classified, but many areas at the edges are not yet classified.

In addition to errors due to segmentation, errors due to classification occur. It has been shown, that the interpretation of object classes strongly depends on the user and that rules have to be defined for each data set. Another error caused by the classification is the wrong classification of objects due to user (click) mistakes. This error can be minimized by a large number of users.

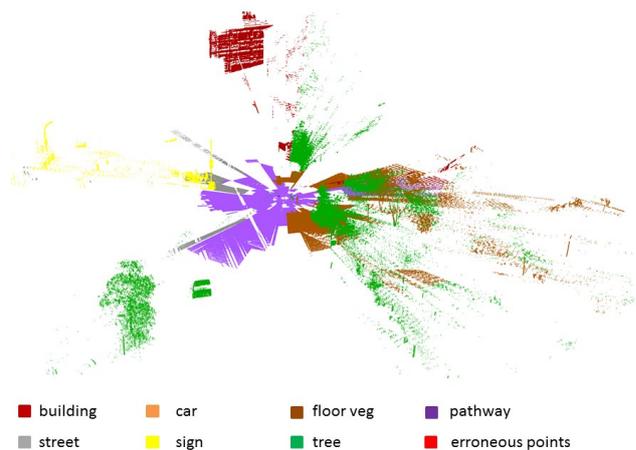


Figure 4. Classified point cloud by the PCCT.

4. IMPLEMENTATIONS AND EXTENTION TO POINTNET

ConvNet PointNet can be used for the identification of individual objects in one point cloud as well as for the semantic classification of scenes. The performance of this ConvNet could be significantly improved by including information across block boundaries (Engelmann et al., 2017 and Qi et al., 2017). Especially the network architecture of ConvNet PointNet without extensions is of interest for the classification of extended and real world point clouds. To our knowledge, PointNet has not yet been applied to complex laser scanning point clouds by now. To investigate the performance of PointNet, a selection of the classified HC point sets was transferred to the block structure of PointNet. The necessary scripts for the data transformation were developed on the basis of the utilities modules. These modules were also designed to duplicate point clouds for training, so that from 7 million points up to 50 million duplicated points can be generated. In addition to the question of how the basic network architecture deals with real laser scanners point clouds, the influence of erroneous points in the point clouds on the classification performance will be investigated. Erroneous points are points caused by sensor technology on one hand, and on the other hand by objects that change their shape and location during recording. Erroneous points due to geometric changes in the object space are unavoidable, especially when measuring outdoors pedestrians, animals and cars causing this kind of error. Due to the size of the point segments, these errors are comparable to the classification of objects. Errors caused by sensors and the measurement setup occur as multipath effects, comet's tail, unfavorable reflections or wrong measurements. These errors are much harder to classify, because they are described only by very few points and occur very irregularly. Most of the available ConvNets do not consider this class because they are designed for synthetic point clouds.

class	precision	recall	IoU
building	0,41	0,60	0,33
car	0,00	0,00	0,00
floor veg	0,31	1,00	0,31
pathway	0,50	0,30	0,24
tree	0,49	1,00	0,49
erroneous points	0,00	0,00	0,00
all classes	0,38	0,60	0,31

Table 2. Results investigation using point clouds with erroneous points. Using the parameters of the precision, recall and IoU.

In order to investigate these hypotheses, different point cloud sets with and without errors were fed into PointNet by using the default settings for iterations (50) and batch sizes (24). The results in the tables 2 and 3 show the parameters for precision, recall and IoU, which are obtained by a classification of five or six classes. These can be directly compared with other investigations. It can be seen that with the laser scanner data a performance of 48 % (IoU) (Qi et al., 2017a), which is based on photogrammetric point clouds, is not obtained for the data set with erroneous points (31 %, IoU). If the identical point cloud only without the class erroneous points is given to the network a better classification result can be obtained with an average of 46% (IoU). An influence of erroneous points in point clouds on classification tasks can be expected, based on these results. This influence needs to be verified by further data and other class compositions. These observations relate to single blocks and

will be extended to the entire point cloud in the upcoming investigation.

class	precision	recall	IoU
building	0,39	0,72	0,34
car	0,43	0,60	0,33
floor veg	0,42	0,83	0,42
pathway	0,46	0,90	0,44
tree	0,49	0,89	0,46
all classes	0,45	0,86	0,42

Table 3. Results investigation using point clouds without erroneous points. Using the parameters of the precision, recall and IoU.

The recall parameter is used to recognize that the assignment to a class depends strongly on the kind of class. In other words, one class can be learned better than the other. This can be observed, for example, at the low represented class car, which has a low recall value.

5. CONCLUSION AND OUTLOOK

The first core task for using ConvNet to classify measured point clouds is to provide a sufficient number of diverse and accurate training point clouds. With the PCCT it is possible to produce these training point clouds efficiently. Even if there need to be done some improves to increase the quality of the PCCT outcome, the PCCT is an import and user friendly tool. In the next PCCT version the segmentation will be improve by using images in various distances.

The second important task is to convert the point clouds from the common surveying formats into a format in which points can be processed with ConvNets without loss of information. A process for this transformation was developed based on PointNet and its utilities. It could be shown that this influence is significant and needs to be further investigated.

REFERENCES

- Barnefske, E., Sternberg, H., 2019. Generation of Training Data for 3D Point Cloud Classification by CNN. *FIG Working Week 2019*, April 22.-26., Vietnam.
- Engelmann, F., Kontogiannia, T., Hermans, A., Leibe, B., 2017. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 716-124.
- Felzenszwalb, P.F., Huttenlocher, D.P., 2004. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2, 167–181.
- Gehring, J., Hebel, M., Arens, M., Stilla, U., 2017. 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*, IV-1/W1. 107–114.
- Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for Autonomous Driving? The KITTI and Vision Benchmark and Suite. *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 3354–3361.

- Girshick, R., 2015. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 1440–1448.
- Grilli, E., Menna, F., Remondino, R., 2018. A review of point clouds segmentation and classification algorithms, *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 339-344.
- Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2017. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 91–98.
- Maturana, D., Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. *International Conference on Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ*, 922–928.
- Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3d classification and segmentation. *Computer Vision and Pattern Recognition (CVPR), IEEE*, 77-85.
- Qi, C.R., Yi, L., Su, H., Guibas, L. J., 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in neural information processing systems*, 2017, 5099-5108
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.
- Rosebrock, A., 2015a. Zero-parameter, automatic Canny edge detection with Python and OpenCV Tutorials, <https://www.pyimagesearch.com/2015/04/06/zero-parameter-automatic-canny-edge-detection-with-python-and-opencv/>, visited 15.01.2019.
- Rosebrock, A., 2015b. Watershed OpenCV in Image Processing, Tutorials, <https://www.pyimagesearch.com/2015/11/02/watershed-opencv/>, visited 15.01.2019.
- Wirges, S., Fischer, T., Stiller, C., Frias, J.B., 2018. Object detection and classification in occupancy grid maps using deep convolutional networks. *21st International Conference on Intelligent Transportation Systems (ITSC)*, 3530-3535