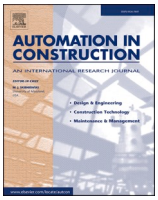
Automation in Construction 138 (2022) 104250

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon



Automatic region-growing system for the segmentation of large point clouds

F. Poux [a,*,1], C. Mattes [b,1], Z. Selman [b,1], L. Kobbelt [b,1]

[a] Geomatics Unit, University of Liège (ULiege), Belgium
[b] Visual Computing Institute, RWTH Aachen University, Germany

ARTICLE INFO

Keywords:
3D point cloud
Segmentation
Region-growing
RANSAC
Unsupervised clustering

ABSTRACT

This article describes a complete unsupervised system for the segmentation of massive 3D point clouds. Our system bridges the missing components that permit to go from 99% automation to 100% automation for the construction industry. It scales up to billions of 3D points and targets a generic low-level grouping of planar regions usable by a wide range of applications. Furthermore, we introduce a hierarchical multi-level segment definition to cope with potential variations in high-level object definitions. The approach first leverages planar predominance in scenes through a normal-based region growing. Then, for usability and simplicity, we designed an automatic heuristic to determine without user supervision three RANSAC-inspired parameters. These are the distance threshold for the region growing, the threshold for the minimum number of points needed to form a valid planar region, and the decision criterion for adding points to a region. Our experiments are conducted on 3D scans of complex buildings to test the robustness of the "one-click" method in varying scenarios. Labelled and instantiated point clouds from different sensors and platforms (depth sensor, terrestrial laser scanner, hand-held laser scanner, mobile mapping system), in different environments (indoor, outdoor, buildings) and with different objects of interests (AEC-related, BIM-related, navigation-related) are provided as a new extensive test-bench. The current implementation processes ten million points per minutes on a single thread CPU configuration. Moreover, the resulting segments are tested for the high-level task of semantic segmentation over 14 classes, to achieve an F1-score of 90+ averaged over all datasets while reducing the training phase to a fraction of state of the art point-based deep learning methods. We provide this baseline along with six new open-access datasets with 300+ million hand-labelled and instantiated 3D points at: https://www.graphics.rwth-aachen.de/project/45/.

1. Introduction

The need to extract structure and knowledge from raw point cloud data is actively driving academic and industrial research. Due to the massive amounts of raw data lacking a structure, many applications require automated pre-processes which can speed up and make existing frameworks more reliable.

Point cloud segmentation is a core component that facilitates the separation of spatial-spectral attributes into their individual constituents. The cardinal motivations for point cloud segmentation are threefold: firstly, it provides end-users with the flexibility to efficiently access and manipulate 3D scenes through higher-level primitives (segments). Secondly, it creates a compact representation of the data wherein all subsequent processing can be done at a region level instead of the individual point level, resulting in potentially significant computational gains. Finally, it gives the ability to extract relationships between neighbourhoods, graphs and topology, which is non-existent in unordered point-based datasets and thus effectively enables the transition from sub-symbolic to symbolic 3D data analysis.

Semantically-rich point cloud datasets can then be used in scenarios such as Building Information Modelling (BIM) reconstruction [1], in Architecture, Engineering, and Construction (AEC) [2], Facility Management (FM) [3] or even automated semantic-based navigation [4].

For these reasons, segmentation is predominantly employed as a preprocessing step to annotate, consolidate, analyse, classify, categorize, extract and abstract information from point cloud data.

* Corresponding author.
E-mail address: fpoux@uliege.be (F. Poux).
[1] All authors contributed equally to this work.

https://doi.org/10.1016/j.autcon.2022.104250
Received 20 July 2021; Received in revised form 5 March 2022; Accepted 5 April 2022
Available online 21 April 2022

However, the biggest challenge lies in overfitting segmentation pipelines to one particular application or process through hand-tuned parameters, which implicitly adds supervision in "unsupervised" schemes. This paper tackles this challenge explicitly by providing a simple and efficient point cloud segmentation system that is fully unsupervised.

We first investigate an objective solution for arbitrary 3D point clouds, transparent enough to be usable on different datasets and within different application domains. Our method considers the Gestalt theory, which states that the whole is more than the sum of its parts and that relationships between the parts can yield new properties/features. We want to leverage the predisposition of the human visual system to group sets of elements. We propose to structure a point cloud in Connected Elements as defined in [5], primarily motivated by the limitations of point-based approaches, where the amount of data, the redundancy, and the absence of relationships among points are significant performance issues. The solution proposes a region growing approach inspired by the "Efficient RANSAC for Point-Cloud Shape Detection" method proposed in [6]. It leverages only X, Y, Z coordinates, uses the same probabilistic paradigm and aim at an industry-ready solution for processing real-world scale datasets.

To cope with the sometimes unintuitive and tedious tuning of parameters and thresholds, we provide an automatic heuristic that permits a "one-click" automatic segmentation. To assess the possibilities provided by the clustering scheme, we present our results on several real-world ground truth datasets. These come from various platforms, sensors and with varying characteristics (density, resolution, precision, noise, occlusion) obtained from terrestrial laser scanners, mobile laser-scanners and passive sensors using photogrammetry, structure from motion and dense-matching. Finally, we provide new open-access dataset(s) with a baseline for scientific benchmarking for the tasks of segmentation, instance segmentation and semantic segmentation. Indeed, in recent years, we noticed a massive shift in the literature from traditional classification methodologies to machine learning. There is now an interest in developing deep neural networks that will hopefully unseat tree-based algorithms from their reigning status as the best learners on tabular data. Specifically, for semantic segmentation workflows, recent architectures such as PointNet++ [7], Superpoint graphs [8], KPConv [9], Minkowski Engine [10] or RandLA-Net [11] are very promising, and have the potential to become central methodologies for specific applications. However, these supervised learning approaches suffer from the need of having specific application-based labelled datasets, which is a current major issue with point cloud data which demands open datasets initiatives such as in [12]. Moreover, the heterogeneity and class variation in which objects are found for example within Cultural Heritage and the difficulty gathering enough training data limit the current application of Deep Learning architectures. Motivated by these strong factors, we provide an unsupervised (in the sense there is no need for training data or human intervention) system that permit to obtain robust segments, simply, in a short time frame and on low-cost infrastructures.

Briefly, this paper makes the following three main contributions:

- a new planar-based region growing segmentation system for massive 3D point clouds;
- An unsupervised heuristic to automatically set optimal parameters;
- A substantial open-source labelled point cloud dataset incl. Instances and classes.

The remainder of this paper is structured, as follows: Section 2 briefly reviews recent related works dealing with point cloud segmentation. Section 3 gives the details of the proposed region-growing segmentation and parameters determination. In Section 4, we present the experimental setup including several indoor datasets, where we let parameters vary and provide details on the implementation. In Section 5, we study the impact of these experiments on the results and analyse the performance of the approach. In Section 6, we discuss our findings and highlight limitations as well as future research directions.

2. Related work

Point Cloud « pure » segmentation algorithms are mainly based on strict hand-crafted features from geometric constraints, heuristics and rules. The main process aims at grouping raw 3D points into unique regions. Those regions correspond to specific structures or objects in one scene with a certain degree of representativity (an object of interest can be "over-segmented" or "under-segmented"). Since no supervised prior knowledge is required, the delivered results have no "strong" semantic information, which is the task of classification also known as semantic segmentation or labelling. However, to reduce the calculation cost, a frequently used strategy is to over-segment a raw point cloud into small regions before applying computationally expensive algorithms that benefit from the new grouping. Voxels can be regarded as the simplest over-segmentation structures, extended to supervoxels which can largely reduce the data volume of a raw point cloud at the cost of some information loss and minimal overlapping.

We propose to categorize the existing segmentation approaches into four major groups as illustrated in Fig. 1.

This categorization sort point-based, edge-based, region-based and energy-based methods (i.e. probability distributions over point data), each holding a wide variety of tactic such as histogram thresholding [13], unsupervised clustering (k-means [14], fuzzy clustering [15], mean-shift [16], graph-based [17]), over-segmentation [5] or gradient method [18]. We encourage the reader to study the exhaustive recent reviews [19–22] for an in-depth understanding of each non-highlighted possibilities and extension to semantic segmentation workflows. For the sake of conciseness and to keep the focus on the content of the paper, we

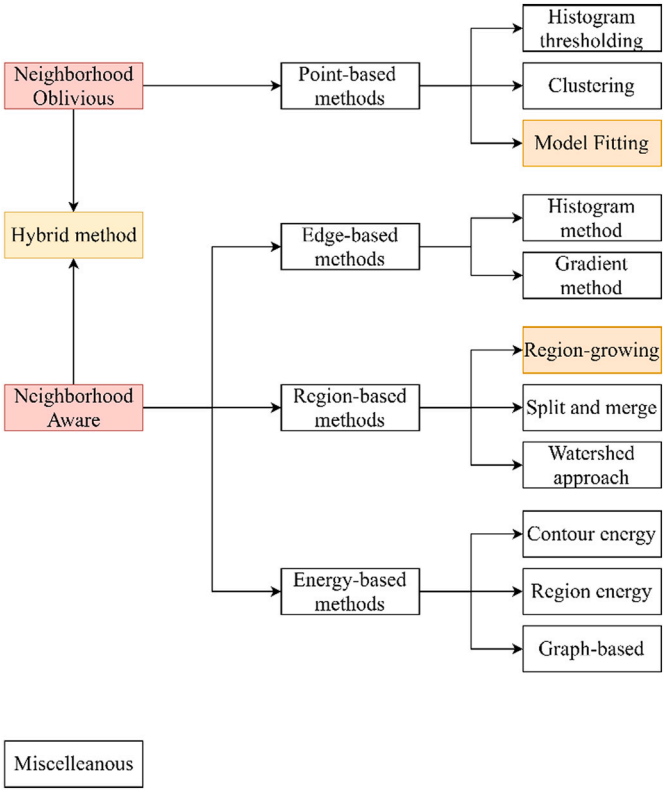


Fig. 1. Taxonomy of existing segmentation approaches. Neighbourhood oblivious are methods that essentially do not uses direct information from a local neighbourhood to create segments, whereas neighbourhood aware build on it.

will specifically provide details of the region growing and model-fitting literature by building on the highly recommended [20] review.

2.1. Region growing

Region growing is a classical point cloud segmentation method which is still widely used today. It is initially derived from images at the pixel level (2D), emerged in 2.5D LiDAR data (possible raster projection), and is now employed at the point level (3D) or the voxel level (3D). 3D region growing comprises two main steps: the selection of seed points or seed units and then the region growing driven by some principles to obtain groups of points following some similarity measure. This technique has been applied in the segmentation of planar building structures in [23,24] or more recently in [25,26]. However, we note that the validation was made on low density dataset or with point cloud never exceeding more than some millions of points, which doesn't presume the complexity of large-scale datasets.

A region growing approach demand the definition of three crucial factors: a seed point selection, a growth unit and a criterion for assessing the similarity between points.

For the seed point selection, since many region growing algorithms aim at a planar segmentation, a usual practice is to fit a plane to a certain point and its neighbours first, and then choose the point with the minimum residual to the fitting plane as a seed point [27]. The residual is usually estimated by the distance between one point and its fitting plane (or the curvature of the point [28]).

For the growth unit factor, there are usually three strategies: (1) single points, (2) region units (voxel grids or octree structures), and (3) hybrid units. Selecting single points as region units was the primary approach in the early stages [27]. However, for massive point clouds, point-wise calculation is time-consuming. To reduce the data volume of the raw point cloud and improve calculation efficiency, e.g., neighbourhood search with a kD-tree in raw data [23], the region unit is an alternative idea of direct points in 3D region growing. In a point cloud scene, the number of voxelized units is smaller than the number of points. In this way, the region growing process can be accelerated significantly. Guided by this strategy, Deschaud et al. [23] presented a voxel-based region growing algorithm to improve efficiency by replacing points with voxels during the region growing procedure. Vo et al. [25] proposed an adaptive octree-based region growing algorithm for fast surface patch segmentation by incrementally grouping adjacent voxels with a similar saliency feature. As a balance of accuracy and efficiency, hybrid units were also proposed and tested by several studies. For example, Xiao et al. [24] combined single points with subwindows as growth units to detect planes. Dong et al. [26] utilized a hybrid region growing algorithm, based on units of both single points and supervoxels, to realize coarse segmentation before global energy optimization.

For the criteria factor, geometric features, such as Euclidean distance or normal vectors, proved efficient. For example, Tovari et al. [29] applied normal vectors, the distance of the neighbouring points to the adjusting plane, and the distance between the current point and candidate points as the criteria for merging a point to a seed region that was randomly picked from the dataset after manually filtering areas near edges. Of course, inaccurate estimations of the normals and curvatures of points near region boundaries are main problematics for these approaches.

Non-universality is a non-trivial problem for region growing [25]. The accuracy of these algorithms relies on the growth criteria and locations of the seeds, which should be predefined and adjusted for different datasets. In addition, these implementations are computationally intensive and may require a reduction in data volume for a trade-off between accuracy and efficiency. Furthermore, we note that all the region growing approaches only work on small scale datasets with a few million points, without a large variability in testing. Also, the definition of the parameters which drive the underlying approaches is supervised with prior knowledge. It essentially limits the generalization of such approaches.

2.2. Model fitting

The core idea of model fitting is to match a point cloud or some subset to different primitives (geometric shapes). It is mainly used as a shape detection or shape extraction method such as in [30]. However, as artificial objects are presents in BIM, AEC, indoor and outdoor scenes are heavily constituted of an assembly of geometric shapes/models such as planes, spheres, cylinders. Therefore, model fitting can be regarded as a segmentation approach to cluster depending on the detected fitted models. The most widely used model-fitting methods are built on two well-known concepts, (1) the Hough Transform (HT) and (2) the RANdom SAmple Consensus (RANSAC).

(1) HT is initially a feature detection technique for digital image processing. It was first presented in [31] for line detection in 2D images. There are three main steps in HT: mapping every sample (e.g., pixels in 2D images and 3D points in point clouds) of the original space into a discretized parameter space; laying an accumulator with a cell array on the parameter space and then, for each input sample, casting a vote for the basic geometric element of which they are inliers in the parameter space; and selecting the cell with the local maximal score, of which parameter coordinates are used to represent a geometric segment in original space. The most basic version of HT is the Generalized Hough Transform (GHT), also called the Standard Hough Transform (SHT), which is introduced in [32]. GHT uses an angle-radius parameterization instead of the original slope-intercept form, in order to avoid the infinite slope problem and simplify the computation. First introduced on a 3D grid, angle-radius parameterization can also be extended into 3D space, and thus can be used in 3D feature detection and regular geometric structure segmentation, as presented in Eq. 1:

$$\rho = xcos(\theta)sin(\phi) + ysin(\theta)sin(\phi) + zcos(\phi) \quad (1)$$

where x, y, and z are corresponding coordinates of a 3D sample (e.g., one specific point from the whole point cloud), and θ and ϕ are polar coordinates of the average vector of the plane, which includes the 3D sample.

One of the significant disadvantages of GHT is the lack of boundaries in the parameter space, which leads to high memory consumption and long calculation time. Therefore, some studies have been conducted to improve the performance of HT by reducing the cost of the voting process. Such algorithms can be found in the review, including Probabilistic Hough transform (PHT), Adaptive probabilistic Hough transform (APHT), Progressive Probabilistic Hough Transform (PPHT), Randomized Hough Transform, and Kernel-based Hough Transform (KHT). In addition to computational costs, choosing a proper accumulator representation is also a way to optimize HT performance as discussed in the review [21].

As with region growing in the 3D field, planes are the most frequent research objects in HT-based segmentation [22,33,34]. In addition to planes, other basic geometric primitives can also be segmented by HT. For example, Rabbani et al. [27] used a Hough-based method to detect cylinders in point clouds, similar to plane detection. In addition, a comprehensive introduction to sphere recognition based on HT methods is presented in [35].

To evaluate different HT algorithms on point clouds, Borrmann et al. [34] compared improved HT algorithms. They concluded that RHT was the best one for point cloud segmentation at that time, due to its high efficiency. Limberger et al. [33] extended KHT to 3D space and proved that 3D KHT performed better than previous HT techniques, including RHT, for plane detection.

2.3. Model-fitting: RANSAC

The well-known RANSAC technique is another popular model-fitting

method first introduced in 1981 by Fischler and Bolles in [36]. High-quality reviews about general RANSAC-based methods have been published in [21], and provide necessary references. While overlapping with these, we clarify the base principle. The RANSAC-based algorithm has two main phases: first, it generates a hypothesis from random samples (hypothesis generation), and secondly, it verifies it to the data (hypothesis evaluation/model verification). Before the first step, as it is the case of HT-based methods, models must be manually defined or selected. Depending on the structure of 3D scenes, these models are generally picked among linear algebraic formulations for planes, spheres, or other geometric primitives.

In the first step of hypothesis generation, RANSAC randomly chooses N sample points and estimates a set of model parameters using those sample points. For example, if the given model is a plane, then N =3, since 3 non-collinear points are needed to determine a plane. The plane model can be represented by Eq. 2:

$$aX + bY + cZ + d = 0 \tag{2}$$

where $[a, b, c, d]^T$ is the parameter set to be estimated.

In hypothesis evaluation, RANSAC chooses the most probable hypothesis from all estimated parameter sets by solving the selection problem as an optimization problem [37] Eq. 3:

$$\widehat{M} = \min_{M} \left\{ \sum_{d \in D} \text{Loss}(\text{Err}(d, M)) \right\} \tag{3}$$

where D is data, Loss represents a loss function, and Err is an error function such as the Euclidean distance.

What is appealing with this sampling approach, is that it does not require complex optimization or high memory resources. Compared to HT methods, the efficiency and the percentage of successfully detected objects are two main advantages for RANSAC in 3D point cloud segmentation. Moreover, RANSAC algorithms have the ability to process data with a high amount of noise, even considering outliers. Analogically to the HT, RANSAC is used in plane-based region-growing segmentation, such as building facades [38], building roofs [39], and indoor scenes [40]. In some fields there is demand for the segmentation of more complex structures than planes. Schnabel et al. [6] proposed an automatic RANSAC-based algorithm framework to detect basic geometric shapes in unorganized point clouds. Those shapes include not only planes but also spheres, cylinders, cones, and tori.

RANSAC is a nondeterministic algorithm, and thus its main shortcoming is its spurious surface: the probability exists that models detected by RANSAC-based algorithm do not exist in reality. To overcome the adverse effect of RANSAC in point cloud segmentation, Li et al. [41] proposed an improved RANSAC method based on NDT cells, also to avoid the spurious surface problem.

Many improved algorithms based on RANSAC have emerged over the past decades to improve its efficiency, accuracy and robustness further as described in [21]. We specifically note the Schnabel et al. approach [6] currently acts as an admitted state of the art resource.

From these targeted related works, it appears that region growing based on normals and RANSAC model fitting are an excellent inspiration to obtain relevant results. However, one has to consider the following points strongly:

- the definition of parameters for the region growing approach;
- the performances and scalability;
- the level of supervision necessary
- the generalization to complex scenes with non-geometrical features

3. Method

Our method considers real-world scans as input and aims at providing a segmented cloud accompanied by an optional neighbourhood graph, as illustrated in Fig. 2. The segmented cloud outputs contain a new scalar field which holds the index of the planar region for each point (X Y Z region).

3.1. Normal estimation

To develop a robust normal-based region growing approach, one needs to extract low-noise normals. Thus, the first step of our approach is to estimate a normal per point. We start by constructing a kD-tree spatial structure of the point cloud to permit quick nearest neighbour queries for each point. A radius search is used to obtain a locally representative subset of the point-cloud.

We then use a principal component analysis (PCA) as in Section 3.1.2 to estimate a local tangent plane and thus we extract an unoriented normal for the point.

If additional knowledge about the position of the scanner or a previous estimate of oriented normals is available (E.g. from terrestrial laser scanners in .e57), we can obtain oriented normals by flipping the computed normals toward the original normals or the scanner position respectively. Oriented normals, however, are not a hard requirement for our method as we will provide simple variants to our main technique that allow for the use of unoriented normals. This permits at a generalizable approach that works also on unordered point sets (E.g., from photogrammetric processes or after a unification process or for datasets from mobile laser scanners as provided in the datasets.

3.1.1. Importance of the neighbourhood radius

To evaluate the dependency of runtime and accuracy from the normal estimation, we run a series of experiments where we uniformly sample points in a 10 × 10 × 0.01 unit volume (meters), which is our probabilistic model of a noisy point cloud with local normal orientation N =(0, 0, ±1).

As shown in Table 1, the size of this radius has a direct impact on the quality of the normals as well as the runtime of the normal estimation. While small values will result in a superior runtime, the resulting normals will be dominated by noise to the point of being unable to capture any useful information as the radius approaches the noise level of the

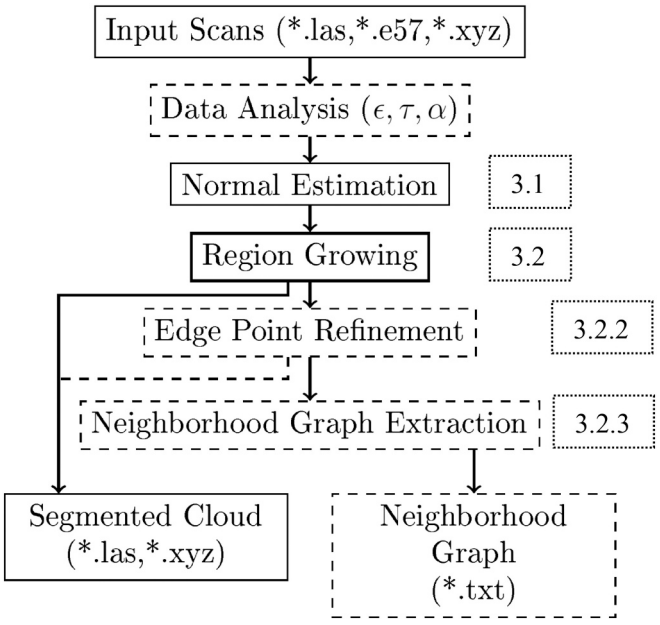


Fig. 2. Workflow of our method, optional parts are represented as dashed boxes. The neighbourhood graph is described as an adjacency list that holds for each region, the regions it is linked to by edges. It is readable directly as an input to classical graph libraries such as NetworkX (Python) or Graphs.jl (Julia) or iGraph (C).

Table 1
Impact of the neighbourhood size ϵ on the runtime t and the average normal deviation $1 - n \cdot \overline{n_0}$. Tested on a point cloud with one million points uniformly sampled from a 10 × 10 × 0.01 volume.

ϵ	t	$1 - n \cdot \overline{n_0}$
0.005	1.19 s	0.81
0.01	2.24 s	0.40
0.02	2.32 s	$1.5 \cdot 10^{-2}$
0.03	2.65 s	$1.7 \cdot 10^{-3}$
0.05	3.38 s	$1.88 \cdot 10^{-4}$
0.1	5.48 s	$1.12 \cdot 10^{-5}$

point cloud. Large values, in turn, will not only immensely increase the runtime of the normal estimation, but will also cause the normals near sharp features in the point cloud to be smoothed out over a larger radius, thus decreasing the quality of the normal estimate and effectively removing small geometric features.

3.1.2. Alternative methods

The main limitation of using PCA for the normal estimation is that we approximate the local neighbourhood of every point in the point cloud with just a single tangent plane. While this approach delivers high quality normals for any point far away from sharp features, any points inside a range of the chosen radius around an edge or corner will be assigned a biased normal (Fig. 3), because two or more planes of the original geometry have to be approximated by a single plane.

While methods that robustly estimate normals yet successfully replicate sharp features in the point cloud are available (e.g. [42]), these come with a substantial runtime overhead (10× for the implementation provided by the authors of [42] as compared to our own PCA-based implementation in Julia).

As later shown in Section 5.3, [42] would thus completely dominate the runtime of our method (in which normal estimation makes up most of the runtime already) which prompted us to rely on the PCA-based approach instead.

3.2. Plane-based segmentation

Once each point is equipped with a normal estimate, we apply a segmentation approach to obtain consistent planar regions from the point cloud. The region-growing method (Subsection 3.2.1) starts by repeatedly selecting a random point that is not yet assigned to a region from the point cloud, and then determine the region it belongs to. If the number of points in the computed region exceeds a heuristic threshold τ (Subsection 3.3.2), it is accepted as a significant region. This process stops once the probability that all planes in the dataset are found reaches a threshold of 99%.

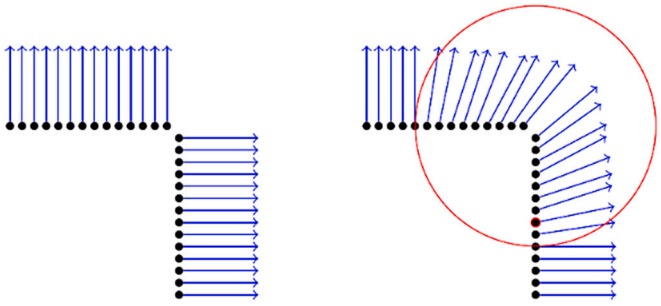


Fig. 3. True surface normal (left) and normal artefacts created by PCA-based normal estimation (right). Normals within an ϵ radius around a sharp edge (red circle) will smoothly blend between the two planes around the edge. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2.1. Growing of an individual region

Starting from a single seed point, growing a region is driven by two index sets *R*(egion) and *F*(ront). As *R* contains the indices of all points we currently consider part of the region, *F* only contains indices of those points that were added to *R* in the last iteration (Fig. 4). Additionally, we keep an estimate of the normal n of the plane, as well as its center of mass c, as they define a plane P.

The algorithm (Algorithm 1) starts by adding the seed point to both *R* and *F*, and initializing n and c with the PCA normal and position of the seed point. In every step of the growing procedure, all the points inside a k-neighbourhood around any of the points in *F* are determined. Of the not yet assigned points in these neighbourhoods, we add those whose normal differs by at most α from n, and whose position has a distance of at most 3ϵ from the PCA-plane to the region *R*. They also replace the previous points in *F*. The maximum width of the extracted region is chosen as 3ϵ to reduce unnecessary over-segmentation in accordance to the prior work of Schnabel et al. [6].

When working on point clouds with unoriented normals, the normal threshold is weakened to only reject points where both, the normal of the point and its flipped counterpart differ by an angle larger than α from n.

To reduce the impact of the seed point selection on the extracted region, the plane estimate is refitted to *R* at exponentially increasing intervals using PCA on the point positions. After each refitting, the points that now fail to fulfil the normal and distance criteria for the new plane estimate are discarded from the region.

As mentioned in Section 3.2, the termination of the region growing procedure is, similarly to the RANSAC-based approach by Schnabel et al. [6], informed by an estimate that all relevant regions have been considered.

Algorithm 1. Algorithm for expanding a region from a single seed point.

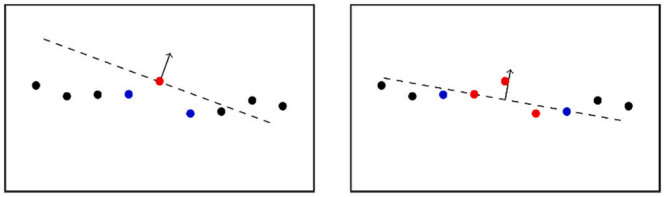


Fig. 4. Expanding a region (*R* in red, *F* in blue): While the noisy normals cause a large error between the estimated and the actual planar area in the beginning (left), the refitting in later steps (right) causes the plane estimate to approach the correct solution rapidly. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Input: point cloud (P, N), seed index i
Output: expanded region R
$c, n \leftarrow p_i, n_i$;
$R \leftarrow \{i\}$;
$F \leftarrow \{i\}$;
$s_{next} \leftarrow s_0$;
while $F \neq \emptyset$ **do**
  $C \leftarrow \bigcup_{j \in F} \{k \mid \|p_k - p_j\|_2 \leq \epsilon\} \setminus R$; /* sped up by kd-tree */
  $C \leftarrow \{j \in C \mid \arccos(n_j^T \cdot n)| \leq \alpha\}$; /* normal criterion */
  $C \leftarrow \{j \in C \mid (p_j - c)^T \cdot n| < 3\epsilon\}$; /* distance criterion */
  $R \leftarrow R \cup C$;
  $F \leftarrow C$;
  if $|R| \geq s_{next}$ **then**
    $c, n \leftarrow$ least-squares plane through R; /* PCA */
    $R \leftarrow \{j \in R \mid \arccos(n_j^T \cdot n)| \leq \alpha\}$; /* normal criterion */
    $R \leftarrow \{j \in R \mid (p_j - c)^T \cdot n| < 3\epsilon\}$; /* distance criterion */
    $s_{next} \leftarrow s_f s_{next}$; /* schedule next refit */
  end
end
return R;

Based on the assumption that the primitive Ψ consisting of g points, can be reconstructed from any sample set of size k denoting the minimal point set required to define a shape candidate within its N points (k being a fixed number dependent on the type of the primitive, $k = 3$ for planes), Schnabel et al., give the probability to randomly select this primitive as [6]:

$$P_\Psi(g) = \frac{\binom{g}{k}}{\binom{N}{k}} \approx \left(\frac{g}{N}\right)^k \tag{4}$$

The probability of finding a region of size g within s trials is thus given as:

$$P_\Psi(g, s) = 1 - (1 - P_\Psi(g))^s \tag{5}$$

Due to the regular refitting of the plane estimate that is part of our approach, we can robustly construct a region from just a single sample point (i.e. $k = 1$) instead of 3 as proposed by [6]. From the equations above, we can see that the number of regions that have to be considered for the segmentation grows exponentially in the number k of sample points required to identify those regions.

3.2.2. Edge-point refinement

The edge-point refinement step compensates the limitations of PCA-based normal estimation. Indeed, due to the nature of the PCA used for normal estimation, normals of hard edges in the point data will be smoothed out. When applying the fast region growing method to a point cloud with PCA-derived normals, the points close to hard edges will therefore usually not be part of any region (unless α is set to at least half the edge angle).

The edge-point refinement permits to correct this effect by reconsidering the assignment of formerly unassigned points in the cloud. For every unassigned point, all regions that appear in the neighbourhood of the point will be considered. Of all found regions, we assign the point to the one region it is found closest to, provided it falls into an 3ϵ-band around the region. So, compared to the original region growing approach, we effectively relinquish the normal angle criterion while keeping the plane distance criterion in effect (Figs. 5, 6, 12, 19, 21, 24).

This refinement step, while fast and easy to implement, permits to obtain a segmentation of good quality even around sharp features, without the need to rely on computationally expensive methods to obtain artefact-free normals, such as the one presented by Li et al. [42].

3.2.3. Neighbourhood relationship extraction

It may be desirable to not only extract planar regions from a point cloud, but also a graph of neighbouring relations. Indeed, this permits to obtain a coarse multi-level topology estimate based on the relations between region entities.

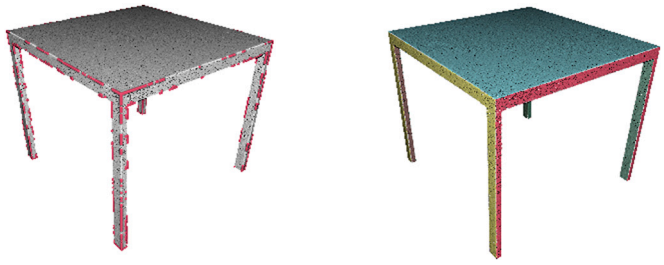


Fig. 5. Illustration of the edge-point refinement step on a synthetic table dataset. On the left, the points (exaggerated) in red are subject of the refinement (0.7% of the dataset), to obtain the final segments (0% of non-attributed points remaining). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Two regions A and B are neighbours if there exist two points $p \in A$ and $q \in B$ such that q is in the ϵ-neighbourhood of p (or vice versa).

3.3. Parameter estimation

Similarly to the shape detection scheme by Schnabel et al. [6] our method relies on three parameters that have to be tuned under consideration of the dataset and the desired quality constraints. These parameters are the neighbourhood radius ϵ, the minimum number of points needed to form a valid planar region τ, and α, the angle decision criterion for adding points to a region. Choosing these parameters optimally however, does not only prove difficult for novice users but can also cause the need for extended experimentation when more advanced users encounter a new dataset or datasets from an unfamiliar scanner. Thus, we provide heuristics for these parameters that can serve as a guideline to the user while requiring little additional computation time.

3.3.1. (Estimating) ϵ

For the parameter ϵ, which will be used for the normal estimation and as the distance threshold for the region growing, a radius that is well above the magnitude of the noise in the dataset is required.

We recall that we are using a PCA to compute the normals of the data points. Part of this PCA is an Eigenanalysis of the neighbourhood of the point, after which the Eigenvector that corresponds to the smallest Eigenvalue becomes the (unoriented) normal of the point. The results of the Eigenanalysis, however, encode additional information about the considered neighbourhood. We can use the ratio between the second largest λ_2 and the smallest Eigenvalue λ_1 as a measure for the planarity of the neighbourhood and thus the reliability of the normal estimate.

To estimate ϵ, we find the smallest neighbourhood radius such that $\frac{\lambda_2}{\lambda_1} \geq 3$ for every point and set ϵ as the median of these values, as we assume that roughly 50% of the points in the point cloud are located on a planar region (as seen in Fig. 7, the actual choice of the percentage is usually noncritical, as for most datasets the median will be part of a large plateau of similar radii). This is supported by uniform density of the point clouds, which is achieved by grid filtering on the high-density raw data.

The influence of ϵ is illustrated on a synthetic dataset in Fig. 8. On the bottom left, unsupervised results ($\epsilon = 2.479$, $\tau = 53$, $\alpha = 29°$), on the bottom right, setting the radius three times higher ($\epsilon = 7.437$). We can see that it mainly impacts the grouping of small region to their adjacent larger counterpart, resulting in fewer segments but potentially losing a finer decomposition into low variance planar regions.

3.3.2. (Estimating) τ

The parameter τ, which is used as a threshold for the minimum number of points needed to support the hypothesis of a planar region,

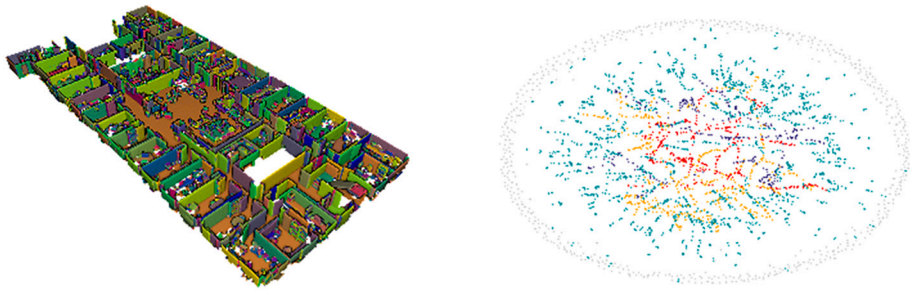


Fig. 6. A segmented indoor point cloud (left) and its neighbouring segments relationships translated into a graph (right). The red nodes have a high centrality and likely refer to central elements (ground, ceiling, walls, table).

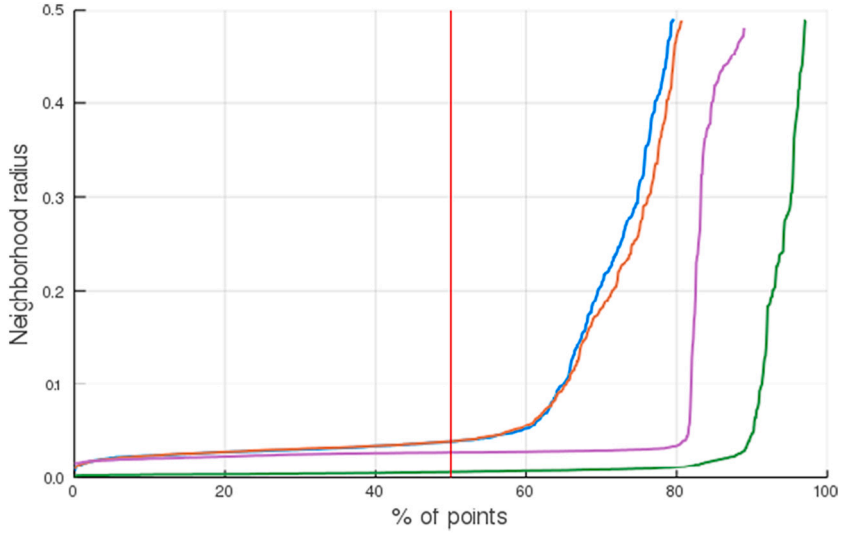


Fig. 7. Required neighbourhood radius to satisfy $\frac{\lambda_2}{\lambda_1} \geq 3$ for the tested datasets. Each dataset is represented by a different colour. The cut-off choice for ϵ (vertical line) is rather lenient, as the median is usually located in a plateau-like interval.

contains more of a user preference than the value for ϵ (e.g. users might want regions above a specific surface area primarily). Rather than completely fixing a value from an analysis of the dataset as we have done for ϵ, we can thus only give some guidance to the user with regards to choosing τ.

As the basis for this analysis, we recall that ϵ is chosen to be well above the magnitude of the noise in the dataset. Thus, the minimal area of a region should be in the order of magnitude of a disk with radius ϵ, as this prevents regions assembled purely from noise data from being accepted.

Since the area of a region is not clearly defined (and even defining it as, for example, the area of the alpha shape of the planar projection of the points, is expensive to compute), we use a threshold number of points τ as stand-in. For this, we again analyse the neighbourhoods of the points in our dataset, this time counting the number of points inside an ϵ-sphere around each point. We then select τ_0 as the median of these point counts.

Should the user have a preference on the minimal surface area A of a region, we can incorporate this by setting $\tau = \frac{A}{\pi\epsilon^2} \cdot \tau_0$. Otherwise, we set $\tau = \tau_0$. Fig. 9 permits to visually assess the influence of τ in the constitution of regions compared to Fig. 8, bottom left. We notice how a higher value for τ favor larger planar regions.

3.3.3. *(Estimating)* α

In order to estimate α, which represents the maximum angle deviation between the normal of a point and the normal of a region for the point to be incorporated into that region, we imagine a sphere of radius 2ϵ (as shown in Fig. 10).

The intuition behind choosing a sphere radius of 2ϵ can be explained as follows: Feature points (e.g. a corner) influence their neighbours within a ϵ-radius (cf. PCA-based normal estimation). Thus, points within an ϵ-radius to a sharp feature have potentially unreliable normal estimates.

A point whose ϵ-neighbourhood is not affected by unreliable normal estimates consequently has to lie at least 2ϵ away from a feature. Another interpretation of this 2ϵ safety margin around sharp features is that region with a curvature radius below 2ϵ are potentially washed-out features, but regions with a curvature radius above 2ϵ are likely to represent an extractable segment. Hence, we choose α such that it properly handles the case of a curvature radius of 2ϵ, i.e. making sure that segments of size $\geq\tau$ can be grown on a sphere of radius 2ϵ. This is also illustrated in Fig. 11.

For this in turn it is necessary to choose α in such a way that a region of size τ can be created on the sphere surface with radius 2ϵ such that all its points fit into a cone with opening angle α and the tip coincident with the center of the sphere. This region will thus form a sphere cap with an area of $A = \frac{\pi\epsilon^2\tau}{\tau_0}$ (as per definition of Eq. 2).

Comparing this with the area of a sphere cap $A = 2\pi rh$ we set $r = 2\epsilon$ and solve for h, yielding $h = \frac{\epsilon\tau}{4\tau_0}$. Since $cos(\alpha) = \frac{2\epsilon - h}{2\epsilon}$, we can derive α as $\alpha = arccos\left(1 - \frac{\tau}{8\tau_0}\right)$.

In the case that τ was chosen as $\tau = \tau_0$, the choice of α is thus independent from the actual dataset yielding $\alpha = 29°$.

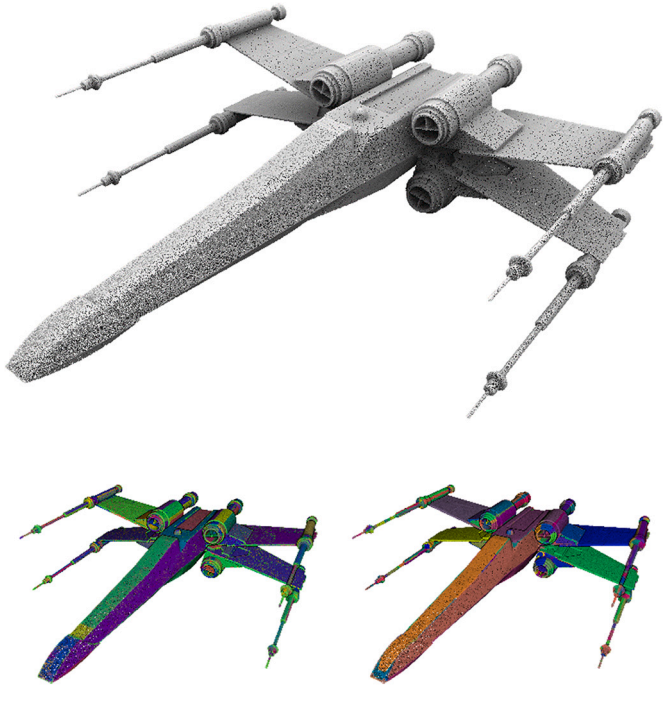


Fig. 8. Influence of parameter ϵ illustrated on a synthetic dataset composed of 1 million points of an X-WING from ShapeNet (Top). On the bottom left, unsupervised results ($\epsilon = 2.479$, $\tau = 53$, $\alpha = 29°$), on the bottom right, setting the radius three times higher ($\epsilon = 7.437$).

Fig. 9. Influence of parameters τ and α illustrated on a synthetic dataset composed of 1 million points of an X-WING from ShapeNet. On the left, setting tau three times higher ($\tau = 159$), on the right, setting the angle two times higher ($\alpha = 58°$).

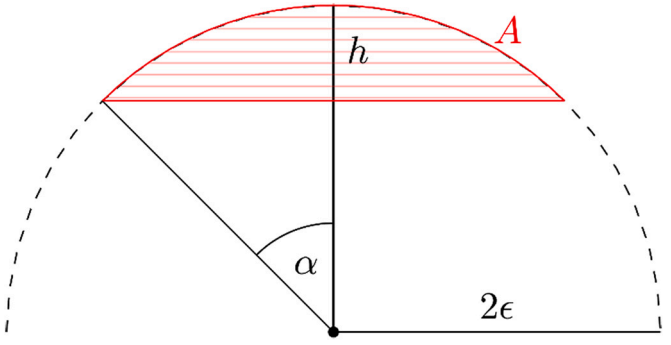


Fig. 10. Cross section of the sphere model used to derive α.

4. Experimental setup

This section aims to detail the specificities linked to the datasets, the metrics, and the parameters used. In turn, it permits establishing a clear protocol for using the datasets as a benchmark basis using novel segmentation architectures.

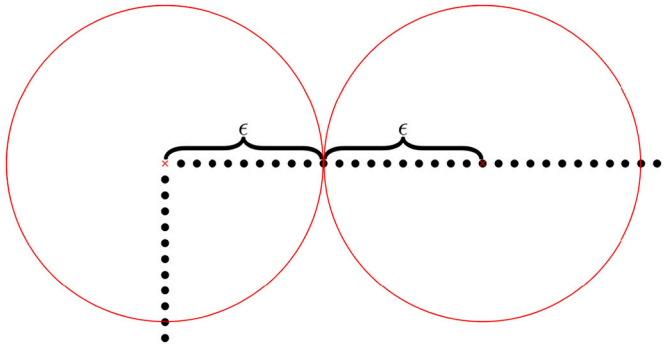


Fig. 11. Two spheres with radius ϵ must be at least 2ϵ apart, to avoid overlap. This way it is possible to avoid washed-out normal that occur close to corners/edges.

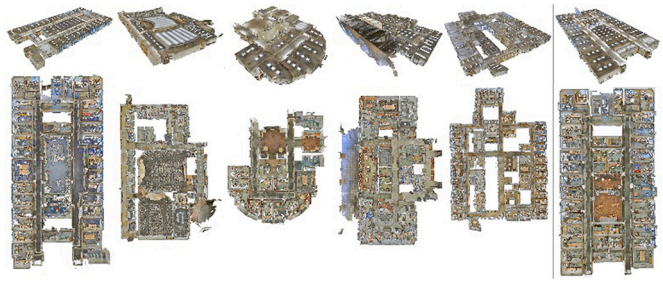


Fig. 12. The different areas of the PCID1.

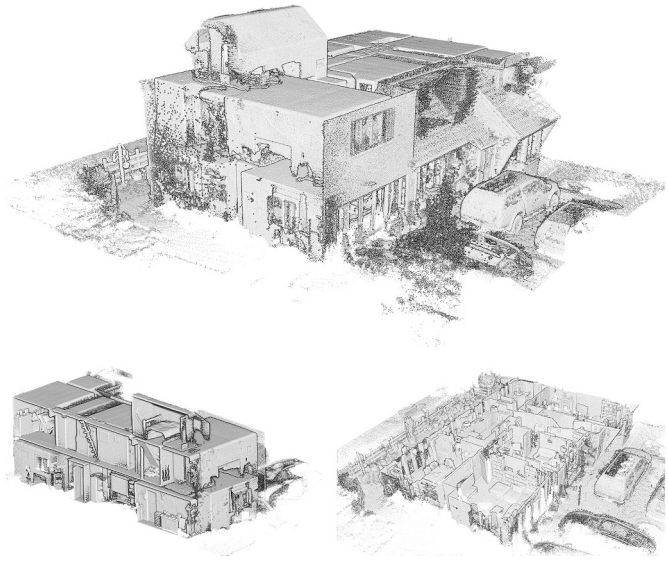


Fig. 13. The different areas of the PCID2. It is a high noise profile point cloud that describes a classical residential house in its full range, including a large number of furniture.

4.1. Point cloud datasets

Four datasets from different sensors and platforms are chosen to gather enough variability found in real-world indoor scenarios: The Leica BLK 360 Terrestrial Laser Scanner (TLS); The Zeb REVO Hand-held Laser Scanner (HHLS); The Matterport depth sensor; The Naavis mobile mapping system (MMS). Table 2 summarises the main characteristics of

Fig. 14. PCID3_1 describe an atypical architecture of an industrial hall with non-linear features.

Fig. 15. PCID3_2 describe an empty office under a construction storey building.

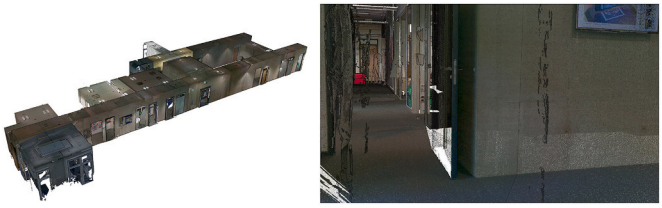


Fig. 16. PCID4 is constituted by a very dense point cloud of the lecture chair at the RWTH Aachen Visual Computing Institute.

each resulting point cloud after registration. They present significant variations in point numbers, density, noise, completeness, attributes and covered areas. The noise profile is estimated by fitting a least-square plane to random floor's planar samples of 2 m^2, averaging and looking at the root mean square error. It varies between 2 mm and 9 mm and is correlated to the point density (the higher the density, the better the fit). Studied point clouds for validation are real-world scenes and hold between 44 and 500 million points.

These dataset from various context are processed to obtain time-wise metrics and visual qualitative assessment as illustrated in Section 5. In the following subpart, we will describe the main characteristic respectively of the Matterport depth sensor (PCID1), the HHLS (PCID2), the MMS (PCID3) and the TLS (PCID4).

4.1.1. Depth sensor Matterport: PCID1

The first dataset used is the S3DIS dataset [43] from the Matterport sensor [44]. It is composed of six storeys areas from different buildings that are each subdivided in a specific number of sub-spaces (i.e. rooms) for a total of 270 sub-spaces. These areas show diverse properties and include 156 offices, 11 conference rooms, 2 auditoriums, 3 lobbies, 3 lounges, 61 hallways, 2 copy rooms, 3 pantries, 1 open space, 19 storage rooms, and 9 restrooms.

One of the areas includes multiple floors, whereas only one is contained in the remaining areas. S3DIS can be described as very representative of indoor building spaces in the United States. The dataset is very noisy, presents imprecise geometries (i.e. geometries that strongly deviate from the real object shapes), clutter and heavy occlusion. We observed several mislabelled points in the ground truth labels and several duplicate points (points where their distance is less than ~9 μm from one another), which add an extra bias. However, it was chosen as it

Fig. 17. PCID4: The planarity dominance of the dataset and uneven sampling due to the sampling BLK unit lead to localized over-segmentation for some walls and doors.

is an important dataset that provides a high variability of scene organization, and it is currently used for benchmarking new deep learning algorithms. It is an interesting opportunity to evaluate our approach's robustness and study the impact of the segmentation and its sturdiness against hefty point cloud artefacts. We remind the readers that the goal is to obtain relevant segments as Connected Elements for high-level usage.

For the automatic classification, we consider 13 classes in the S3DIS dataset, each being decomposed in a fixed number of instances per area used to check for segmentation relevance, as provided in Table 3.

4.1.2. HHLS ZEB REVO gen 1: PCID2

The second dataset is courtesy of GeoSlam, and was acquired by the hand-held laser scanner ZEB REVO (first generation). The subject of the acquisition is a residential house described only through a texture-free point cloud (X, Y, Z only), as illustrated in Fig. 13. This dataset is characterized by a high level of noise, tight spaces, outdoor elements, and household furniture.

The ground truth labelling was made by using semi-automatic techniques described in [45]. F. Poux and A. Kharroubi independently classified it, then merged to limit subjectivity bias. This procedure showed a maximum deviation of 5% in the point classification interpretation. The same technique was used for PCID3 and 4.

Thus, the ground truth data is independent of the proposed segmentation and classification procedures. However, the marking still retains imperfections to permit a qualitative analysis of both the segmentation and supervised classifiers to handle mislabelling cases.

4.1.3. Mms NAAVIS gen 1: PCID3

The PCID3_NAAVIS dataset was acquired by the mobile mapping system Naavis 1 (first generation). It contains X, Y, Z and R, G, B attribute values per points and is decomposed into two areas. The first one represents an industrial hall with complicated non-planar structures

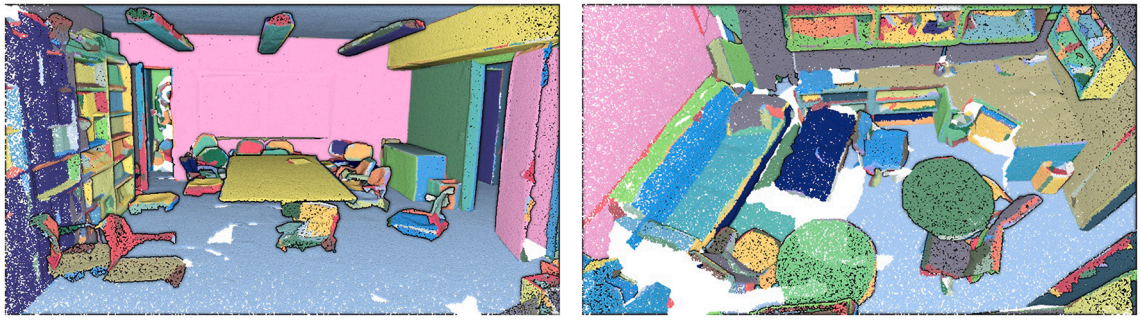


Fig. 18. PCID1_Area_1 Zoom in on two different points of interest. We can note a good planar delineation and natural objects frontier adequations. Additionally, over-segmentation happens almost automatically on any objects composed of multiple planar regions (chairs, lights, fixtures, bookshelves …).

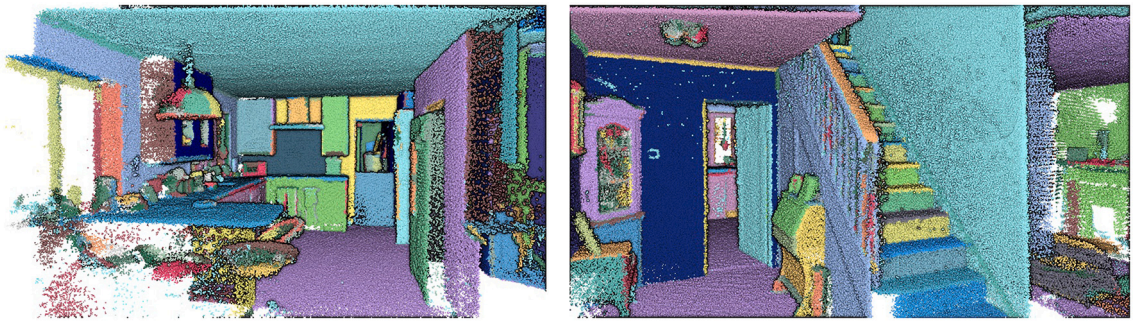


Fig. 19. PCID2_INDOOR We note a good qualitative resistance to high-noise profile point clouds. This permits to find stairs sub-elements, doors, frames, fixtures, and shelves with a high visual robustness. Additionally, we note the good large planar region detection such as walls, ceilings, and floors.

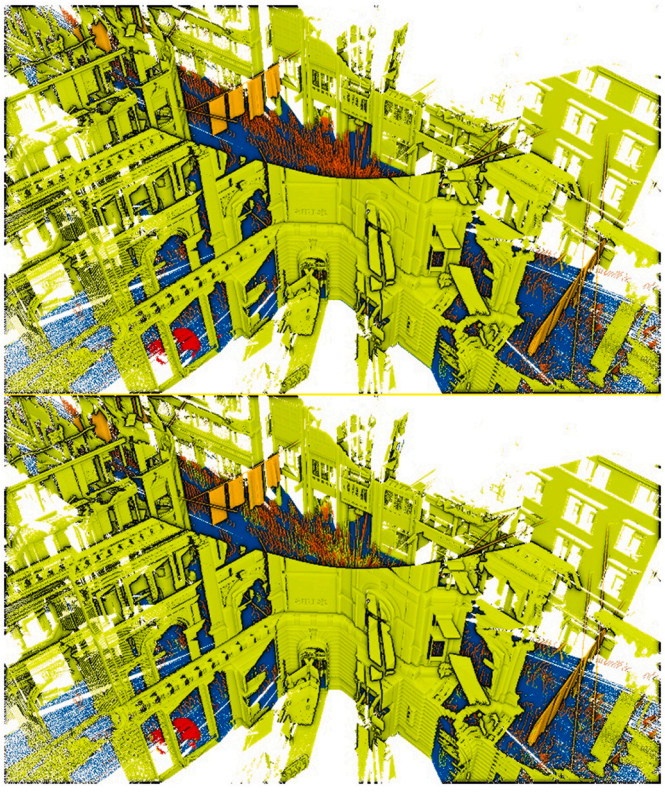


Fig. 20. Top image: Segmentation results. Bottom image: Ground truth. The semantic3d dataset showcase that the errors are often found on the very tiny segments that present a particular local point organization.

(Fig. 14). The scale and the geometries and spatial context between objects are radically different from the other datasets.

The second one is a "clean" indoor level freshly renovated like the Matterport sensor organization (Fig. 15). This dataset has the particularity of presenting large planar surfaces with a low level of occlusion, as the rooms are mainly empty (pre-renovation scenario).

4.1.4. TLS LEICA BLK360: PCID4

The PCID4_RWTH dataset was acquired by the terrestrial laser scanner Leica BLK 360. It represents an indoor typical lecture chair in Germany (Fig. 16). Its specificity lies in a very high point density and low noise level and includes a registration error to specifically study the ability of the approach to handle such cases. The ground truth labelling was made using the same procedure as described above.

4.2. Parameter settings

The determination of the parameters was done following the automatic heuristic procedure described in Section 3. However, to qualitatively assess its reliability, we compared it to hand-tuned values. In most cases, we found the results produced by applying the parameter estimation methods described in Section 3 to be preferable to manually selected and fine-tuned parameter values. We found that we ourselves would often overestimate the ideal values for ϵ and τ to avoid a costly recomputation of normal estimate and segmentation, while lower values for these parameters would afford a better extraction of detailed information from the point cloud.

4.3. Metrics

In our experiment, we provide segmentation metrics (Subsection 4.3.1) to quantify the accuracy and robustness of the unsupervised experiments, and classification metrics (Subsection 4.3.2), to quantify the supervised learning experiments based on the segmentation results.

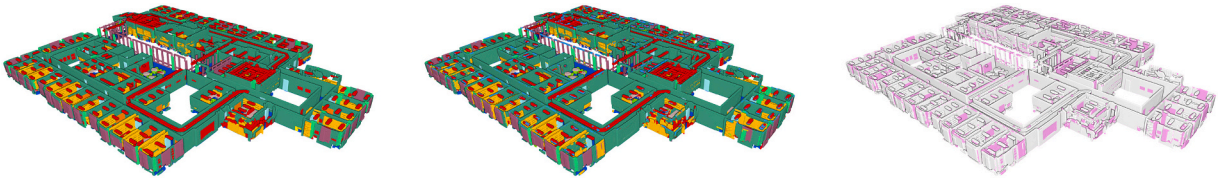


Fig. 21. Qualitative results on the most problematic area of the PCID1 Matterport sensor. From left to right: Ground Truth Point Cloud; Predicted classes; Differences to Ground Truth in rose. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

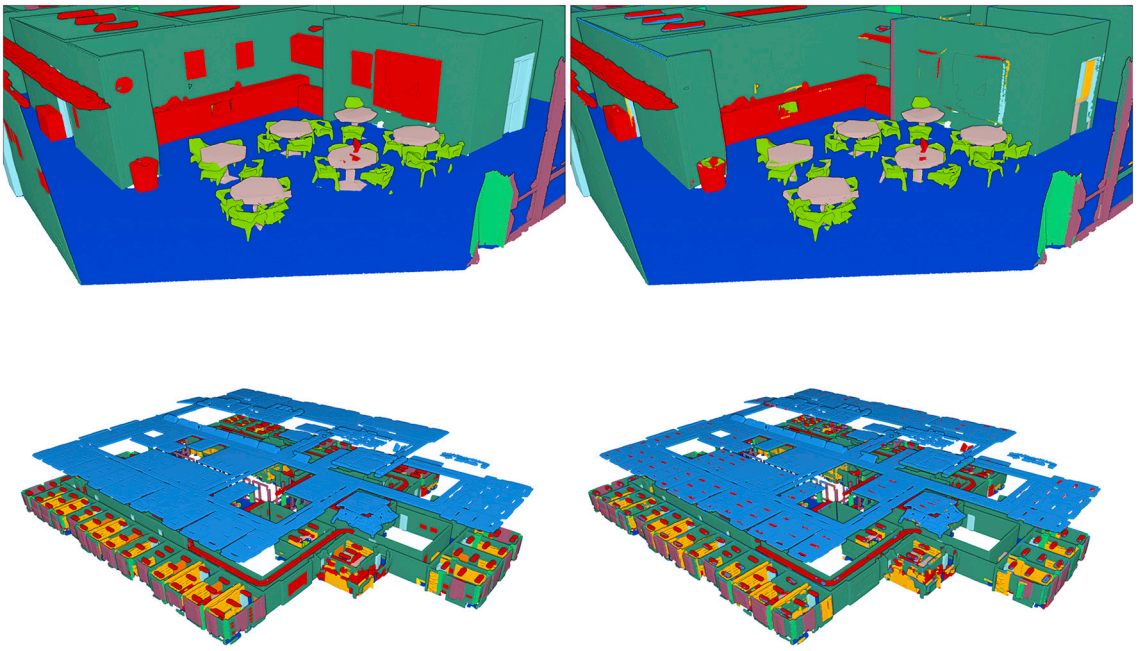


Fig. 22. Zoom in onto problematic cases such as the blackboards hanging on the wall, with a very small normal deviation. Left shows the ground truth, Right the predictions. Colours would help in these scenarios.

4.3.1. Segmentation metrics

To quantify the quality of our segmentation approach, we first determine an "ideal" segmentation from the ground truth data by identifying connected components of points with the same classification. This segmentation is ideal because it contains the minimum number of segments while still allowing a per-segment classification with perfect accuracy.

In practice, however, because our method identifies planar regions instead of more complex objects that will be assigned to a single class in the ground truth (such as a chair or a bookcase), a significant amount of over-segmentation is expected. While strong over-segmentation will impact the performance of the subsequent classification step, it will usually not reduce the quality of the classification. We measure the over-segmentation of a ground truth segment as the number of planar regions for which the current ground truth segment has the largest overlap (measured in points).

A more critical aspect of the segmentation is under-segmentation, i.e., combining multiple ground truth segments into a single extracted segment, as this actively limits the accuracy any per-segment classification can achieve. Similarly, to over-segmentation, the under-segmentation of a planar region is measured as the number of ground truth segments for which the current planar region has the largest overlap.

However, a low amount of under-segmentation does not directly imply a good segmentation quality, as a planar region might still overlap multiple ground truth segments leading to misclassifications, while not making up a large enough fraction of each segment to be counted as under-segmentation. As a more direct measure of the classification quality that we can achieve with a given segmentation, we first assign every planar region to the ground truth segment with the largest overlap. We then define the "sharpness" of the segmentation as the percentage of points where the ground truth segment and the assigned segment of their planar region are identical. The sharpness of the segmentation is thus an upper bound for the accuracy that a subsequent region-based classification can achieve. Given the set of ground truth labels $L = \{0, \ldots, 8\}$ (8 for Semantic3D) and the set of clustered segments $S = \{0, \ldots, N-1\}$ for N segments found by our method. Let l_i/s_i be the label / segment ID for point i. Let v_i be the voted label for point i. Each segment votes for a label by majority such as in Eq. 6:

$$vote(s) = argmax\{li \mid si = s\}, s \in S, \forall i \tag{6}$$

We use arg max to denote the most frequent element of that set. We set $v_i = \text{vote}(s)$, for all $s_i = s$. We define sharpness as follows:

$$\text{sharpness}(c) = \frac{\#\{i \mid vi = ci, ci = c\}}{\#\{i \mid ci = c\}} \tag{7}$$

4.3.2. Semantic segmentation metrics

Existing literature has suggested several quantitative metrics for assessing the semantic segmentation and classification outcomes. We define the metrics regarding the following terms that were extracted from a confusion matrix C of size $n \times n$ (with n the number of labels, and each term denoted c_{ij}):

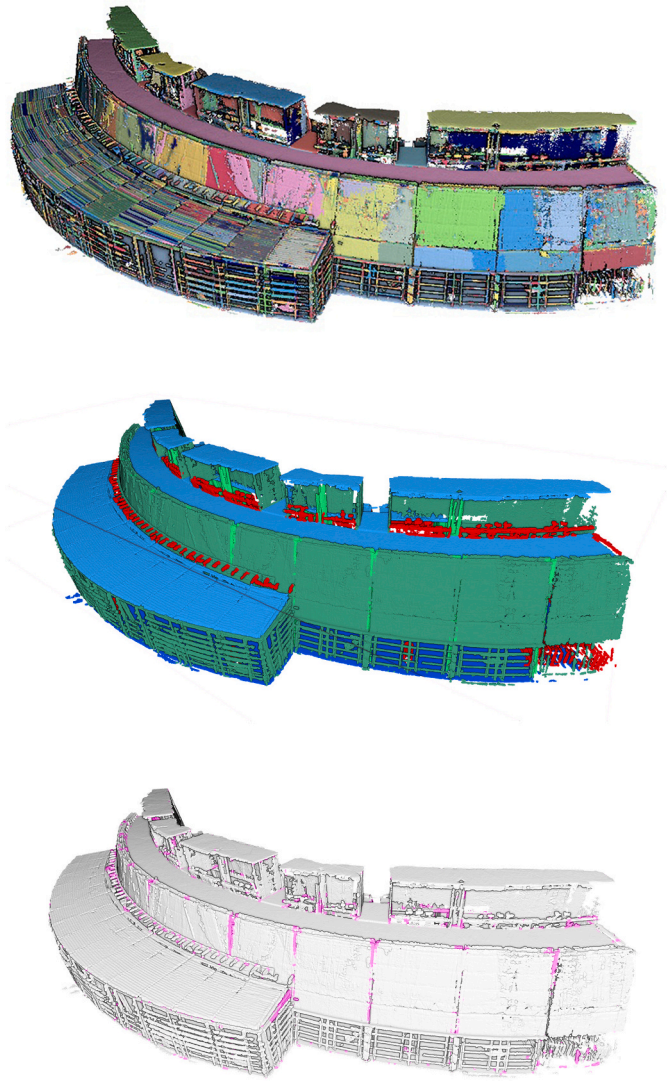


Fig. 23. PCID3 qualitative results illustrated. First, we see the over-segmented point cloud obtained following our methodology, its translation into tabular data, that is fed to the generalized classifier, to obtain the classified point cloud, highlighted for its deviation from the ground truth in rose. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- True Positive (TP): Observation is positive and is predicted to be positive.
- False Negative (FN): Observation is positive but is predicted negative.
- True Negative (TN): Observation is negative and is predicted to be negative.
- False Positive (FP): Observation is negative but is predicted positive.

Subsequently, the Intersection-Over-Union (*IoU*), Precision, Recall and $F_{1-\text{score}}$ metrics are used. Precision is the ability of the classifier not to label as positive a sample that is negative, the recall is intuitively the ability of the classifier to find all of the positive samples. The F1-score can be interpreted as a weighted harmonic mean of the precision and recall, thus giving a good measure of how well the classifier performs. Indeed, global accuracy metrics are not appropriate evaluation measures when class frequencies are unbalanced, which is the case in most scenarios, both in real indoor and outdoor scenes, since the dominant classes bias them.

In general, the Intersection-Over-Union (*IoU*) metric tends to penalize the single instances of bad classification more than the F1-score, even when they can both agree that this one instance is bad. Thus, the *IoU* metric tends to have a "squaring" effect on the errors relative to the F1-score. Henceforth, the F1-score in our experiments gives an indication on the average performance of our proposed classifier, while the *IoU* score measures the worst-case performance.

4.4. Estimated parameter values

The parameters used in our experiments were derived according to Section 3.3. In Table 4, both the values of the parameters ϵ and τ, as well as the amount of time used for their estimation, is shown. Note that in most cases both parameters can be estimated in less than a second and the estimation would beat a cycle of manually choosing a parameter value, running the segmentation and adjusting the parameters based on the result by several orders of magnitude in every case.

5. Results

The experimental protocol described in Section 4 is followed to obtain quantitative and qualitative results over point cloud segmentation (Subsection 5.1) and semantic segmentation (Subsection 5.2).

5.1. Segmentation results

A shortened summary of the over- and under-segmentation metrics, as well as the sharpness, is presented in Table 5. It becomes clear that our method introduces a significant amount of over-segmentation, which is expected as our method only recognizes planar regions while segments in the ground truth dataset can be of any shape (Fig. 18 or in Appendix A).

On the other hand, however, we find that under-segmentation is not a significant issue with our method, as we do not observe any under-segmentation for over 99% of ground truth segments. And even in cases where under-segmentation occurs, this is largely due to features such as boards mounted tightly to walls that are close to impossible to recognize using point cloud geometry alone and would require the inclusion of colour information into the clustering.

Solely for the PCID4 dataset, we also studied the combination of uneven sampling and the fact that almost all points lie on large, planar surfaces (rather than the 50% assumed in Section 3) on any potential increase in over-segmentation of the point cloud. We found that it does not affect the semantic segmentation quality by testing this dataset with an estimate for ϵ based on an assumed inlier percentage of 85% from domain knowledge injection (Illustration in Fig. 17).

As an additional view, we provide an in-depth view of the parmater's computation in Table 6.

Although our method produces reasonably good results on Semantic3D, a significant increase in computation time is noted. Due to each cloud consisting of a single scan-position the point density decreases with distance from the origin. Starting from ~3 mm between scan-lines near the centre, this distance increases to 18 cm on far-away buildings. Operations leveraging neighbourhood information, e.g. normal estimation, require to increase their search radius in accordance with the lowest point-density in the cloud. Searching in a large neighbourhood in a high-density region is bound to increase computation time, as the number of points inside the neighbourhood increases. This is predominantly noticed for normal estimation, a small radius will cause arbitrary normals in low-density areas. However, this limitation is in practice lifted as the datasets that present such extreme density variation are found in structured file formats (E.g. e57), and these datasets usually hold normal information that permit to bypass the normal estimation. The classes present in the Semantic3d dataset are summarized in Table 7.

We present in Table 8 the sharpness results of the segmentation on Semantic3D and illustrated over Fig. 20.

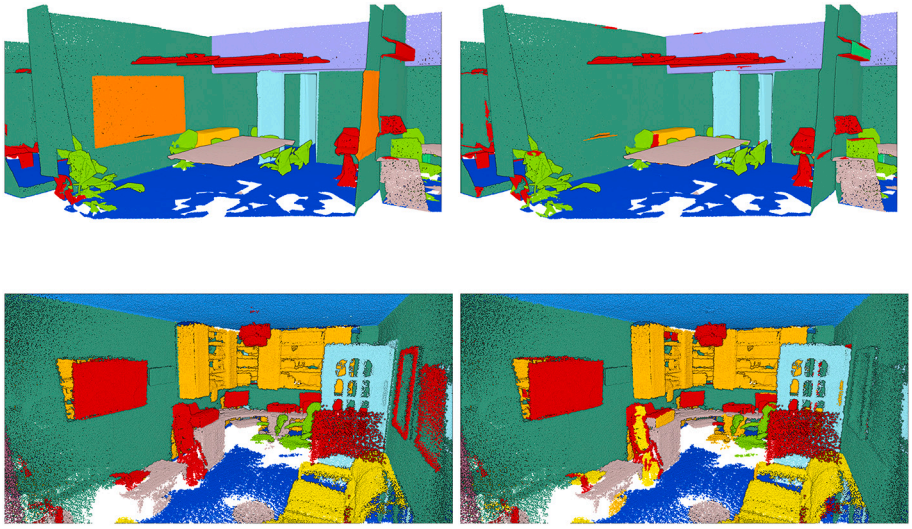


Fig. 24. Qualitative results over PCID1 (top) and PCID2 (bottom). On the left are the Ground Truths; on the right are the Generalized Predictions. We can see that errors are usually spatially contiguous (due to the segmentation), and point toward small segments part of moveable furniture's. In some occurrence, we also find a mislabelling for planar shapes that resemble thus of the "clutter" class, which present a high variability. Also, in PCID1, we see the misclassification of the board hanging on the wall, responsible in the drop of scores.

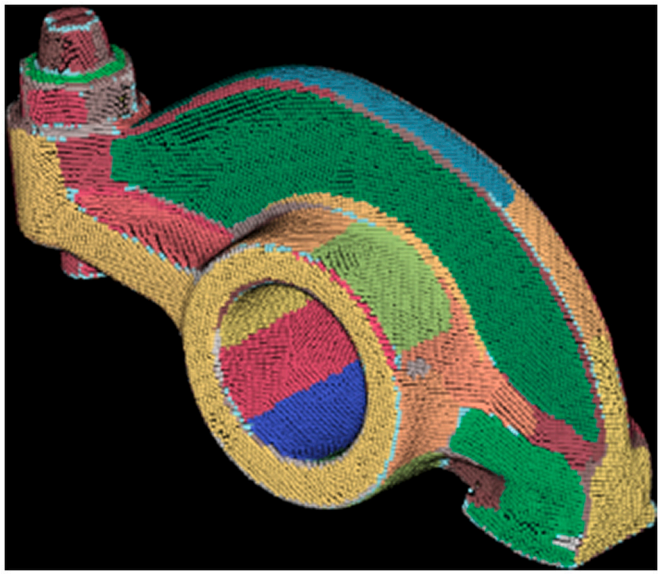


Fig. 25. Segmentation system result on the rocker arm model. The natural borders of finite elements of the model closely fit the natural ones present in the original implementation.

Table 2
Characteristics of the different datasets provided. Each is classified to provide ground truth data for evaluating segmentation and classification performances.

Model	Matterport	Zeb REVO	Naavis 1	Leica BLK
Identifier	PCID1	PCID2	PCID3	PCID4
Areas nb.	6	2	2	1
Type	Depth Sensor	HHLS	MMS	TLS
Points nb (in mil.)	273	45	45	235
Density (pts/cm^2)	3	2	9	40
Classes number	13	20	9	13
Size (.las)	8.86 GB	1.37 GB	1.97 GB	7.46 GB
Attributes used	XYZ only.			
Attributes added	Instance number + Class			
Noise profile	4 mm	9 mm	3 mm	2 mm
Occlusions	Yes	Yes	Yes	Yes
Area	6000 m^2	450 m^2	1500 m^2	635 m^2

Our method produces good results on dominant labels in the data sets. We note a limitation of our approach linked to the assumption of approximate uniform point density. While it tends to improve computation times for several steps, it cannot be generally assumed for arbitrary data sets. We opted against grid filtering the data in Semantic3D to allow for better comparison. Further improvements will take non-uniform density into account, in order to avoid costly neighbourhood queries in high-density regions.

Finally, we provide a comparison to three state of the art Deep Learning architectures in Table 9: KPConv being the current best performer [9], Superpoint graphs [8], and PointNet++ [7].

5.2. Semantic segmentation results

The semantic segmentation step was developed to study the suitability of the segmentation's results and its uplifting effects for high-level instance segmentation frameworks. The classifier's input is a list of segments, each described by 11 simple features (size as the point number, centroid coordinates, main normal directions, normal variance, length, width and height of the segment). The segment's list is then fed to several classifiers: Support Vector Machines, Multi-Layer Perceptron, Naïve Bayes, Neural Networks, and Random Forests. Due to the relatively low number of observations per area and features per segments, we orient our choice toward the best performing supervised approaches as studied in [46]. Indeed, the authors delineate in [46] the dominance of such classifer for tabular datasets, with this low number of features (<50) and low number of records (<1 million). Thus, we present only the results of the Random Forest for their good representativity with low-overfitting capabilities. Indeed, this permits us to get a quantitative sense of the gain one can achieve with a relatively simple workflow and with proven supervised approaches.

First, each area is studied for establishing a model using a 60/40 clear split between training and test data. Then the predictions at the segment level are directly transferred to the point level. This permit first to get a sense of performance on a narrow context (linked to a dataset at hand). Then, we create a generalized model by training only on 50% of all segments from all datasets (287 638 segments) combined with a 5-fold cross-validation to study the generalization potential to other datasets while constraining to a small training dataset. The emphasis in experiments is toward a simple architecture. Therefore, hyper-parameters are fixed for all datasets to "standard" values to emphasize segmentation's results rather than any classifier performances (1000 decision trees where each tree expand until every leaf is pure, each internal node must have at least two samples before it can be split and

Table 3
Classes repartition of the S3DIS dataset (PCID1).

Classes	ceiling	floor	wall	beam	column	Window	door	table	chair	sofa	bookcase	board	Stairs	Clutter
ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Area 1	55	1	234	61	57	29	86	69	155	6	90	27	2	755
Area 2	81	9	283	11	19	8	93	46	545	6	48	17	11	495
Area 3	37	1	159	13	12	8	37	30	67	9	41	12	0	335
Area 4	73	1	280	3	38	40	107	79	159	14	98	10	3	671
Area 5	76	1	344	3	74	52	127	154	258	11	217	42	0	922
Area 6	63	2	247	68	54	31	93	77	179	9	90	29	1	684
Total Num	385	15	1547	159	254	168	543	455	1363	55	584	137	17	3862

Table 4
Estimated parameter values and time (t_ϵ) needed for estimation. τ is the number of points.

Cloud	ϵ (m)	t_ϵ (s)	τ
PCID1_Area_1	0.036	0.416	36
PCID1_Area_2	0.035	0.371	35
PCID1_Area_3	0.034	0.366	34
PCID1_Area_4	0.035	0.365	35
PCID1_Area_5	0.035	0.361	37
PCID1_Area_6	0.036	1.519	37
PCID2_REVO_INDOOR	0.034	3.362	104
PCID2_REVO_OUTDOOR	0.048	1.000	62
PCID3_NAAVIS_1	0.019	0.583	17
PCID3_NAAVIS_2	0.044	0.105	15
PCID4_RWTH_CHAIR	0.007	15.586	29

every leaf must have at least 1 sample that it classifies). In order to clearly study the impact of the segmentation on the results, we addressed under-segmentation by attributing to faulty segments the dominating classes (i.e. the class where the number of points is the highest in one segment), thus potentially predicting on biased segments. In Table 10 we gathered our metrics for a representative area of the PCID1, illustrated in Fig. 22. We already note that one can achieve a high accuracy for planar dominant classes such as ceiling, floor and walls, even with a very limited number of observations. If we take a closer look, we determine minor geometric variation to be the main cause of problematic cases, such as a blackboard hanging on the wall. One would need to use other features such as colour information to be able to detect such entities).

To conduct a consistent evaluation, we first test our supervised approach on each area of each dataset independently, which holds a wide array of rooms with varying size, architectural elements, and problematic cases. On top, the data present non-planar ceiling, stairs, heavy noise, heavy occlusion, false-labelled data, duplicate points, clutter, and non-planar walls. A summary of the results is given in Table 11.

While the data presents several challenges, we obtained resulting F1-scores that globally vary between 88.0% and 98.1%. While planar

Table 7
Classes present in Semantic3d dataset.

Class0	Unclassified
Class1	Man-made terrain
Class2	Natural terrain
Class3	High vegetation
Class4	Low vegetation
Class5	Buildings
Class6	Hard Scape
Class7	Scanning artefacts
Class8	Cars

Table 5
Shortened overview of the over−/under-segmentation metrics. We note the high consistency of the very low under-segmentation results over the different areas through different context. The lowest sharpness score of the Area 5 of PCID_1 is 86.1%, and can be explained by a high misclassification proportion on many occasions, at the borders of large segments. Dense point clouds show results above 96% sharpness, due to a better planar fit.

Dataset	Over-segmentation			Under-segmentation		Sharpness
	Median	75%	Max	99%	Max	
PCID1_Area_1	11	17	3883	1	9	89.92%
PCID1_Area_2	12	20	644	1	13	90.34%
PCID1_Area_3	11	21	417	1	9	91.25%
PCID1_Area_4	9	19	896	1	15	89.86%
PCID1_Area_5	10	18	1525	1	21	86.1%
PCID1_Area_6	11	18	794	1	9	90.61%
PCID2_REVO_INDOOR	5	13	679	1	4	90.42%
PCID2_REVO_OUTDOOR	17	44	436	1	3	96.83%
PCID4_NAAVIS_1	78	186	6770	1	2	98.68%
PCID9_NAAVIS_2	10	19	243	1	10	91.64%
PCID10_RWTH_CHAIR	86	300	14,453	1	4	96.39%

Table 6
Performance metrics in seconds for the Semantic3D data. Contains the different steps from Kd-tree constitution, to the estimation of parameters, the computation of normal and the clustering step. A striking change compared to the performance on PCID is the estimation timing for ε. As opposed to the other data, Semantic3D has no filtering applied. Each cloud represents an unfiltered scan-position, with increasing point density closer to the respective origin. For any given sphere, the number of points inside is higher, closer to the origin. As a result, neighbourhood-based operations such as ε or normal estimation take longer. Estimation of normals is not taken into account for total runtime, as this is not a contribution of our method, and some datasets may already provide them.

Data Set	N	KD-Tree (s)	Est. ε (s)	Est. τ (s)	Normals (min)	Clustering (s)	Total (s)
UB1	27.977 M	8.40	178.716	0.450	323.4	322.2	509.8
UB3	28.059 M	8.30	168.375	0.361	319.1	191.2	387.4
NG1	50.122 M	15.60	219.419	0.425	93.1	312.1	547.5

Table 8
Sharpness results of the segmentation on Semantic3D, numbers in percent. Our method produces good results on dominant labels in the data sets. Less frequent occurring labels suffer from the fact that we favour large regions. The zero for Class 2 indicates that no cluster was correctly assigned this label.

Dataset	Total	Class0	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8		
NG1	99.77	27.71	99.21	N/A	61.01	49.94	99.84	96.44	67.46	88.54		
UB1	99.54	71.27	99.39	58.93	60.06	89.57	79.48	98.16	87.28	93.96		
UB3	99.77	66.65	97.92	79.41	56.53	85.41	99.99	17.31	62.62	89.03		
SG27S5	96.49	68.67	90.88	97.71		99.97	55.1	93.28	51.13		74.62	85.52
SG27S9	99.93	79.02	99.7	97.03		50.3	74.45	99.98	82.47		78.16	84.27
SG28S4	97	67.69	99.53	84.72		44.55	97.65	94.97	20.27		62.69	86.17
SG27S4	98.51	80.9	99.22	83.5		99.96	49.88	74.74	52.04		73.77	93.8
SG27S1	97.92	49.68	88.92	96.8		32.11	28.68	90.88	32.08		31.88	42.08
SG27S2	98.93	16.7	91.42	85.17		29.4	47.6	93.81	34.27		99.89	62.86

Table 9
Comparison to State-of-the-Art Deep Learning architectures for the Semantic3d dataset.

	OA	A_IoU	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8
NG1	98.3%	69.7%	98.9%	N/A	61.0%	49.7%	97.0%	96.2%	67.0%	88.1%
UB1	92.0%	78.2%	98.7%	56.0%	60.1%	89.3%	79.4%	60.7%	86.9%	93.8%
UB3	89.5%	70.6%	97.7%	74.8%	56.5%	84.9%	82.1%	17.3%	62.5%	88.6%
SG27S5	92.6%	75.0%	89.7%	92.2%	59.2%	55.1%	93.3%	51.0%	74.0%	85.3%
SG27S9	97.9%	81.1%	99.7%	95.9%	50.3%	74.1%	85.1%	81.5%	78.1%	84.1%
SG28S4	93.4%	71.5%	95.6%	82.8%	44.5%	85.5%	94.9%	19.9%	62.7%	85.8%
SG27S4	90.1%	74.6%	96.9%	82.2%	73.9%	49.4%	74.7%	52.0%	73.8%	93.7%
SG27S1	90.0%	53.3%	85.6%	83.0%	32.1%	28.5%	90.9%	32.0%	31.9%	42.1%
SG27S2	83.6%	60.0%	89.7%	84.4%	29.4%	47.3%	93.7%	34.2%	38.2%	62.8%
ConvPoint	95.0%	77.7%	95.9%	90.0%	79.1%	70.5%	96.3%	43.3%	56.1%	90.7%
SPGraph	92.9%	76.2%	91.5%	75.6%	78.3%	71.7%	94.4%	56.8%	52.9%	88.4%
PointNet++	85.7%	63.1%	81.9%	78.1%	64.3%	51.7%	75.9%	36.4%	43.7%	72.6%

Table 10
IoU scores comparison for PCID1 Area 5 based on 60/40 split results from the segmentation.

PCID1_A5	Ceil.	Floor	Wall	Beam	Col.	Win.	Door	Table	Chair	Sofa	Book.	Board	Stairs	Clutter	W. Av
(7 million)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
Precision (%)	97.2	98.9	82.8	10.5	82.1	92.4	90.8	85.4	74.8	88.0	88.5	88.1	–	79.2	89.2
Recall (%)	97.9	99.2	97.1	21.4	68.7	55.1	77.6	86.3	89.0	69.8	81.2	2.1	–	65.4	88.9
F1-score (%)	97.6	99.0	89.4	14.1	74.8	69.0	83.7	85.9	81.3	77.9	84.7	4.1	–	71.7	88.1
IoU (%)	95.3	98.1	80.8	7.6	59.7	52.7	72.0	75.3	68.5	63.7	73.5	2.1	–	55.9	78.7

Table 11
F1-score at the point level of the Random Forest classifier based on fully unsupervised results from the segmentation.

F1-score	Ceil.	Floor	Wall	Beam	Col.	Win.	Door	Table	Chair	Sofa	Book.	Board	Stairs	Clutter	F1
60–40-U	0	1	2	3	4	5	6	7	8	9	10	11	12	13	score
PCID1	97.7	97.2	88.3	79.6	79.7	81.9	77.2	87.6	90.7	77.2	84.1	24.2	56.0	76.6	88.6
PCID2	92.8	64.4	92.9	65.5	66.9	86.8	80.4	90.2	89.5	91.4	87.1	–	62.4	84.1	88.6
PCID3	98.8	99.2	96.0	42.8	89.7	48.2	91.7	44.5	–	–	–	–	97.4	82.8	96.7
PCID4	99.0	99.5	95.5	98.5	88.1	–	88.1	97.9	98.6	98.9	–	–	–	92.4	96.1

dominant classes are among the best-recognized classes, the improvement margin is found for the board, sofa, column, and windows classes.

It is interesting to note that while chair variation in orientation, shape, size and distribution is really high, the detection rate is still performing above 83.9%, and is consistent independently of the dataset. Also, the clutter class presents a relatively low score as it contains a large variance in the "objects" it contains, thus providing a great candidate to improve scores at the classification level by refining elements within. It is interesting to note that the segmentation results at this step permit to obtain satisfactory results, independently of the indoor scene and sensor characteristics. However, we note a very low score for the beam detection of Area 5, which is due to the very low number of beam occurrences in this area, thus a biased prediction.

In Table 13, we provide a general quantitative summary per datasets regarding each sensor if we want to obtain a more high-level overview. This confirms the suitability of the segments independently used per sensor to create a classifier that can achieve F1-scores above 88% with relatively low hyperparameter tuning and resources demand. The major let-down concerns PCID 6, where the first floor is mislabelled, but the F1-score presents a good overhaul. The highest score is obtained by PCID4, which is largely explained by the low noise ration of the point cloud, and the sharpness of the segmentation results (96.39%). We generally observe that the metrics follow closely segmentation sharpness results, which is the major influencer on the quality of the downward process.

Then, we created a "generalized" model by training only on 50% of the full number of segments, and controlling that the number of segments per area used doesn't exceed 10% of the area's segment number. On top, we used a 5-fold Cross validation to obtain a good idea about a general performance within a wide array of variations presented in Section 4. To summarise our segment-based Random Forest performances, we present in Table 12 the main results per class, and several screenshots of various prediction's example in Fig. 23 and in the Appendix A.

Table 12
Generalized Random-Forest Classifier performances (in %). We notice that we have a drop overall of 3.4 points, but the model obtain is then able to generalize to many classes and a lot of scenarios with ease and insuring 80% scores, which permits to deploy an automatic labelling service with minimal control afterwards.

Generalized	Ceil.	Floor	Wall	Beam	Col.	Win.	Door	Table	Chair	Sofa	Book.	Board	Stairs	Clutter	F1
PCID1 (%)	96.8	80.9	86.4	71.5	71.7	71.6	74.9	65.1	83.8	67.9	76.2	17.3	52.8	71.6	82.8
PCID2 (%)	92.1	68.2	92.2	34.4	29.2	77.1	77.3	67.2	58.9	75.0	77.0	–	72.4	70.2	85.8
PCID3 (%)	97.9	99.2	94.3	40.4	87.4	38.0	85.0	22.1	–	–	–	–	78.5	72.3	94.3
PCID4 (%)	98.1	99.5	93.6	77.4	81.8	–	79.8	95.3	82.0	85.4	–	–	–	82.2	93.3

First, we notice a drop of 3.4 F1-score points in average to obtain a generalized model. This is mainly explained by the heterogeneity in which the objects in the scene are found. Indeed, we have more variants and thus it impacts extracted features from the segments. Thus, it consolidates the segmentation's accuracy, and highlight its fit to semantic segmentation frameworks.

5.3. Implementation and performance details

The runtimes of the different parts of our algorithm are listed in Table 15. From these timings and the sizes of the used datasets we conclude that for clouds in the order of magnitude of 100 million points, we can expect a clustering performance of around 200K points per second (when averaging over the total runtime, not just the clustering step). To put this into perspective, the largest dataset tested by Schnabel et al. in "Efficient RANSAC for Point-Cloud Shape Detection" contains around 1.9 million points with a runtime of 61.5 seconds (for $\tau = 500$), which equates to roughly 31K points per second.

Mind that the time needed for clustering is not quite linear in the number of points. For the rocker arm model (40K points, Fig. 25), the total runtime of our method is around 0.2 seconds, while Schnabel et al. give an average runtime of 6.5 seconds for their method.

Finally, we implemented and added a performance comparison to the octree-based region growing approach from A. Vo et al. ([25] reported in Table 14:

Even with these numbers, we still see potential to improve the performance of our method: We implemented our method in Julia and focused more on simplicity and legibility than pure performance. Furthermore, the clustering itself does not yet make use of multiple threads.

As an additional remark, we used these following libraries:

- Kd-tree and kNN: NearestNeighbors.jl. However, we developed its support for out-of-core structuration as the point cloud we processed did not fit in the RAM (at the cost of a slight longer processing time)
- PCA: we used the standard library LinearAlgebra.jl, that hold functions to get eigenvalues and eigenvectors
- LasIO and/or Laspy: for reading .las files (point cloud datasets)

6. Discussion

From the detailed analysis provided in Section 5, we first summarise the identified strengths in Subsection 6.1. We then propose the main research directions for future work addressing the limitations in Subsection 6.2.

6.1. Strengths

First, the region growing segmentation is fully unsupervised, which gives a significant edge over supervised approaches or manual parameter iteration. The definition of a fully automatic process to determine optimal parameters with no prior knowledge injection gives a lot of freedom of applicability. Its robustness supports this to varying scenarios and conditions as proven by the wide array of real-scene testbench. It also highlights its potential to generalize with relative ease to scenarios that extend the scope of planar dominant scenes.

Secondly, the presented method is easy to implement. It does not rely on GPUs and mainly leverages the CPU coupled with available RAM. It is crucial for many companies that do not possess high-end servers equipped with expensive GPUs. It is easily deployable on a small infrastructure, without the need to upgrade the server-side. The coherent results are expected in less than 20 minutes for a dataset of 200 million points with a 5 years old laptop (i7-Gen6 CPU and 12 GB of RAM). As it stands, without deep optimizations, it permits offline automatic segmentation and classification, and the data structure provides parallel-computing support. Third, we provide a segment-based Random Forest classifier that delivers state of the art results with minimal training time and an excellent potential to generalize.

One can easily incorporate it in a semantic segmentation production-ready workflow by labelling a small sample of the data. We tested the idea with up to 10% of any dataset as training data giving F1-scores above 90%. We also combined our results with point-based classifiers, as illustrated in [47].

Fourth, there is a low input requirement that only necessitates unordered X, Y, Z datasets. The segmentation provides a complete directed graph of the relations within segments by a kD-tree matching the segment structures presented in Section 3. This information permits reasoning services to use the connectivity information between objects and subspaces for advanced queries using spatial and semantic attributes.

Finally, unsupervised segmentation and supervised classification are easily extensible by limiting over-segmentation. For example, one can differentiate clutter based on connectivity and proximities to enhance the classification further (e.g., clutter on top of a table may be a

Table 13
F1-scores at the point level of the Random Forest classifier per area based on fully unsupervised results from the segmentation.

F1-score	Ceil.	Floor	Wall	Beam	Col.	Win.	Door	Table	Chair	Sofa	Book.	Board	Stairs	Clutter	F1
60–40-U	0	1	2	3	4	5	6	7	8	9	10	11	12	13	score
PCID1_A1	98.8	98.4	86.8	90.1	80.5	80.0	73.8	84.6	83.9	51.4	80.5	33.1	90.8	83.1	88.0
PCID1_A2	97.0	89.2	89.0	85.2	76.8	97.6	67.9	87.6	97.2	86.7	74.2	33.4	92.4	67.2	86.9
PCID1_A3	98.5	99.3	87.0	94.8	77.3	72.8	79.0	88.5	93.1	88.8	89.7	30.4	–	78.0	89.2
PCID1_A4	95.3	98.8	89.4	89.7	82.7	82.5	79.5	86.6	89.9	88.4	85.6	16.3	77.2	71.5	88.4
PCID1_A5	98.6	99.1	90.9	26.0	80.2	81.0	86.4	89.5	90.4	76.5	88.3	12.1	–	79.1	90.8
PCID1_A6	98.1	98.8	86.5	92.1	80.9	77.6	76.5	88.9	90.1	71.6	86.1	19.9	75.3	80.7	88.3
PCID2_I	92.8	64.4	92.9	65.5	66.9	86.8	80.4	90.2	89.5	91.4	87.1	–	62.4	84.1	88.6
PCID3_A1	99.5	99.7	96.2	–	96.0	–	96.2	88.9	–	–	–	–	99.3	79.6	98.1
PCID3_A2	98.1	98.7	95.8	85.6	83.3	96.4	87.1	–	–	–	–	–	95.6	85.9	95.3
PCID4	99.0	99.5	95.5	98.5	88.1	–	88.1	97.9	98.6	98.9	–	–	–	92.4	96.1

Table 14
Results of comparing our approach against the octree-based and point-based segmentation by Vo et al. and Rabanni et al.

	Subtest		60E	30E	15E	10E
Number of points		5,954,336	2,993,506	1,578,713	1,036,185	712,868
Density (points/m2)		3019	1360	630	393	264
Build kd tree (in seconds)	Rabanni et al. (2006)	28.41	1.11	5.63	3.44	1.78
	Ours	2.05	1.62	0.78	0.45	0.29
Compute point's features (in seconds)	Rabanni et al. (2006)	556.39	304.07	156.26	86.45	51.20
	Ours	9.49	4.32	2.59	1.90	1.48
Point-based region growing (in seconds)	Rabanni et al. (2006)	625.83	345.97	177.70	102.12	58.95
	Ours	27.01	14.11	8.16	5.11	4.08
Total segmentation process (in seconds)	Rabanni et al. (2006)	1210.63	66.11	339.58	192.01	111.93
	Ours	38.55	20.05	11.52	7.46	5.85
Comparisons	[25]	37.81	29.96	19.02	10.89	5.75
	Rab/Ours	31.41	3.30	29.48	25.74	19.13
	[25] /Ours	1.02	0.67	0.61	0.69	1.02

Table 15
Performance metrics, in seconds for the different steps from Kd-tree constitution, to the estimation of parameters, the computation of normal and the clustering step.

Data Set	N	Kd-Tree	Est. e	Est. τ	Normals	Clustering	Total
PCID1_Area_1	44.196 M	16.4	0.416	0.002	49.6	162.3	179.1
PCID1_Area_2	47.315 M	10.6	0.371	0.001	77.1	171.4	182.4
PCID1_Area_3	18.662 M	6.2	0.366	0.001	17.6	64.4	71.0
PCID1_Area_4	43.47 M	15.5	0.365	0.001	47.8	157.0	172.9
PCID1_Area_5	78.719 M	25.5	0.361	0.001	106.2	282.8	308.7
PCID1_Area_6	41.353 M	10.4	1.519	0.284	55.9	152.8	165.0
PCID2_1_INDOOR	41.508 M	14.2	3.362	0.005	138.8	166.1	183.7
PCID2_2_OUTDOOR	4.626 M	1.9	1.000	0.003	8.9	21.1	24.0
PCID3_ 1_HALL	44.577 M	14.6	0.583	0.001	54.2	172.4	187.6
PCID3_ 2_STOREY	4.446 M	1.4	0.105	0.001	3.7	14.8	16.3
PCID4	229.582 M	82.9	15.586	0.319	321.3	1091.4	1190.2

computer; clutter linked to the ceiling and in the middle of the room is a light source…). These potentials are addressed as research tracks for future works, as presented in the following subsection.

Down the line, it can be used for many applications, such as extracting the surface of ceilings, walls, or floors if one wants to make digital quotations. It can provide a basis for extracting semantic spaces (sub-spaces) organized regarding their function or to provide a basis for floor plans, and visualization purposes.

6.2. Limitations and research directions

While successful in various tasks, supervised learning requires a large amount of human-labelled data. This learning method tends to produce task-specific, specialized systems that are often brittle outside of the narrow domain they have been trained on [48]. Reducing the number of human-labelled samples or interactions with the world required to learn a task and increasing the out-of-domain robustness is crucial for applications within the construction industry.

While our unsupervised method permits us to create with relative ease and robustness semi-automated labelling workflow toward creating labelled datasets, it currently cannot compare the ability humans have to learn massive amounts of background knowledge about the world task-independent manner. It motivates a form of prediction or reconstruction such as self-supervised learning to "fill in the blanks" by predicting masked or corrupted portions of the data [48], which is frequent in point cloud datasets. Our first results are promising [49]. Still, its extension to better handle uneven density and manage relationships with a hierarchical view of the "objects and segments composing a scene constitute an exciting research direction.

As a worthwhile aim for future work, we consider adding recognition of higher-level primitives (such as spheres, cylinders, tori, etc.) based on the existing planar-based segmentation, such that we obtain a complete alternative to RANSAC-based methods while retaining simplicity and computational efficiency. On top, considering other attributes such as colour or intensity could help our system recover from limitations in detecting elements undistinguishable through geometric changes alone.

7. Conclusions

In this paper, we have outlined a region-growing based system for the segmentation of large point clouds. We have demonstrated the simplicity of the method, both in the implementation as well as in the application. The system depends on only three relatively intuitive parameters, which were previously proposed by Schnabel et al., and for which we presented a fully automatic heuristic determination that can derive these parameters in negligible time frames. This approach allows inexperienced end users to perform segmentation, without requiring domain knowledge. To verify the validity of our segmentation results and demonstrate the applicability to the semantic segmentation of massive point clouds provided by a wide range of acquisition methods, we applied a supervised classifier in an object-based fashion. Our results indicate capability to deal with very large datasets and good results in sharpness of segmentation. This permits us to obtain a baseline for emerging deep learning models that learn efficiently on tabular data. We provide a selection of new open-access datasets ranging from just below 4.5 M points up to almost 230 M points.

Funding

This work was funded by the European Regional Development Fund within the "Terra Mosana" project under the funding code EMR10.

Declaration of Competing Interest

None.

Appendix A

Below are some additional snapshots of the results of the segmentation process.

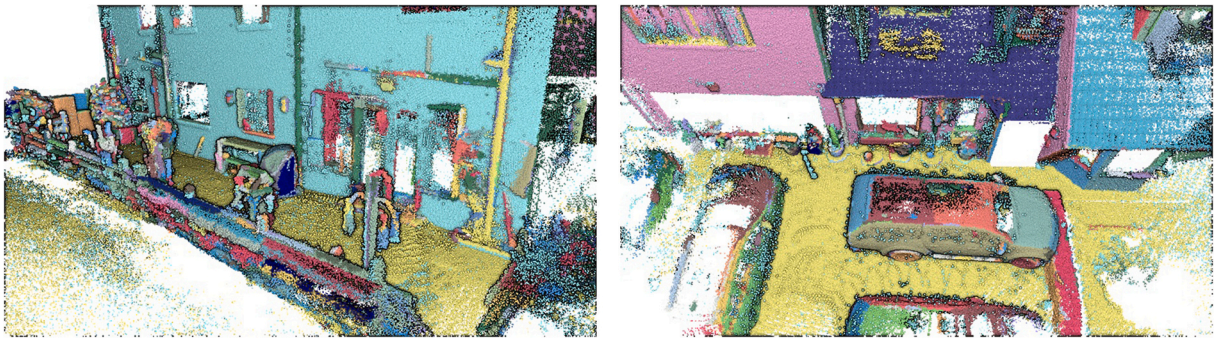


Fig. A.1. PCID2_OUTDOOR Results of the region growing process. We note a high level of over-segmentation, especially for "linear" shapes which presents uneven sampling.

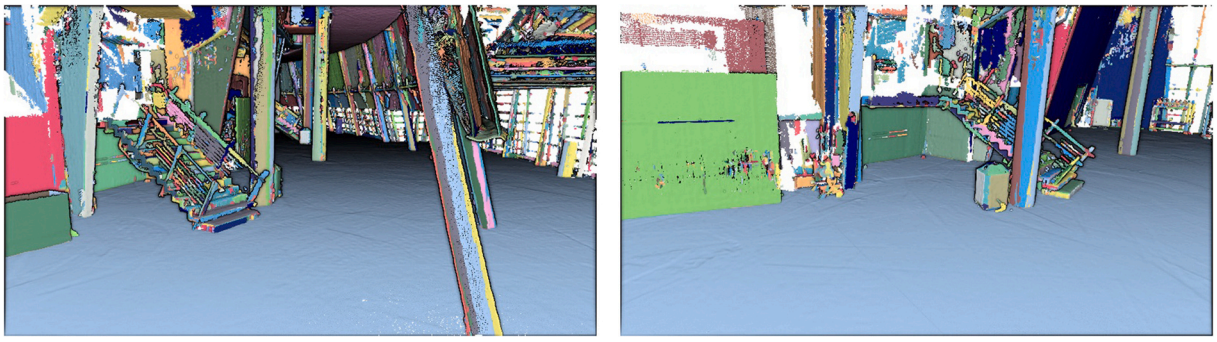


Fig. A.2. PCID3_1. The over-segmentation clearly delineates the cylindrical shapes such as the columns holding the roofing structure. This is a great illustration of research directions toward multi-modal primitive support.

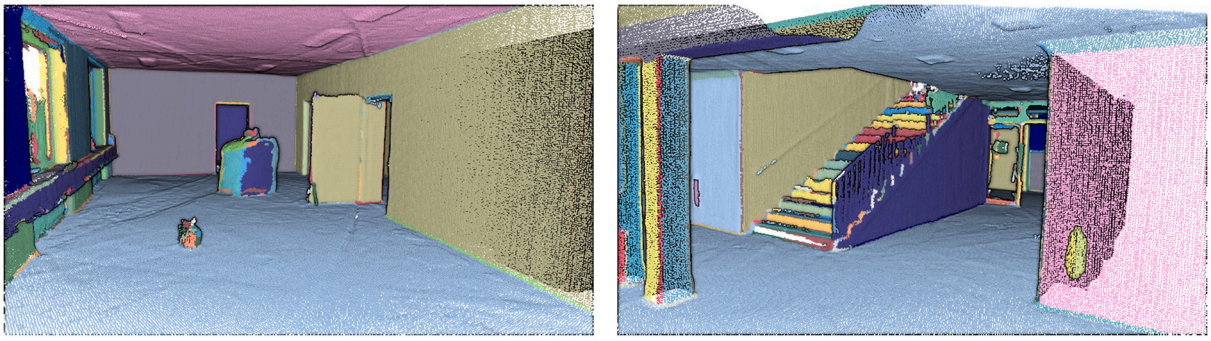


Fig. A.3. PCID3_2 This dataset shows an artefact at the edges, where the distinction between wall and ceiling is not clear, a region is found.

Below are the results of the object-based semantic segmentation workflow based on the 11 features extracted from the segments.

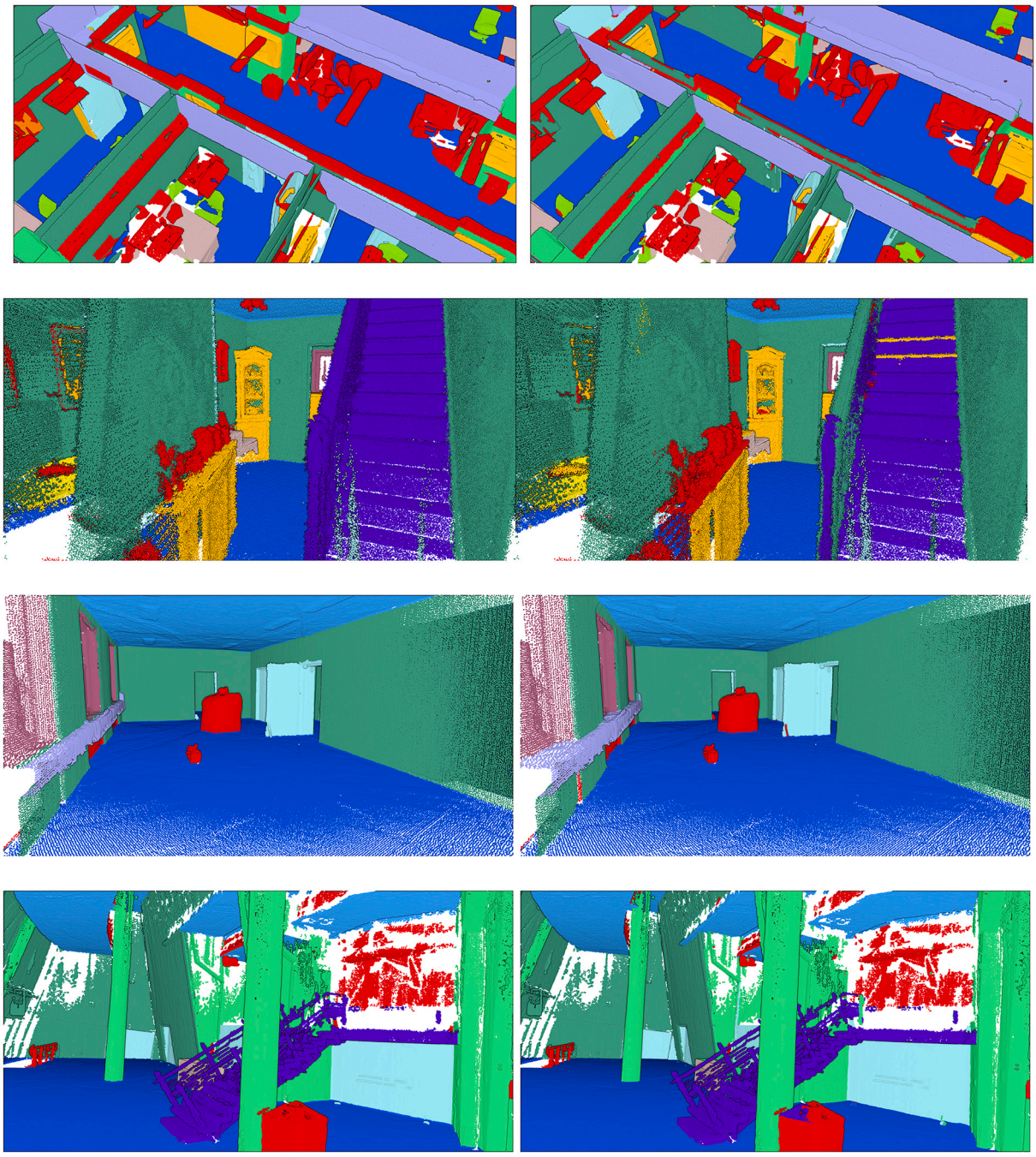


Fig. A.4. Qualitative results over PCID1, PCID2 and PCID3. On the left are the Ground Truths, on the right are the Generalized Predictions.

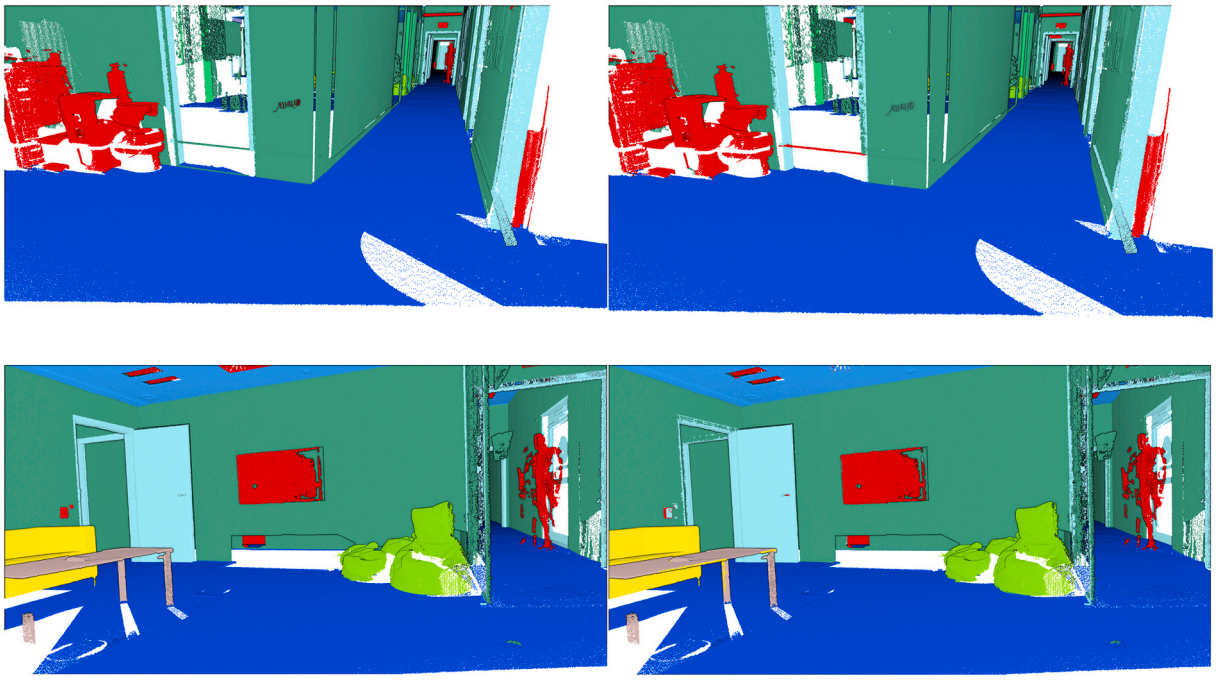


Fig. A.5. Qualitative results over PCID4. On the left are the Ground Truths, on the right are the Generalized Predictions.

In order to have a complete view of the performances of our segmentation system, we provide a microbenchmark in Table A.1. The configuration on which we perform this is an Intel i7–3770 CPU @ 3.40 GHz and with 12.0 GB RAM.

Table A.1
Microbenchmark of parameter estimation for PCID. Total number of runs per parameter: $N = 100$. Timings in seconds. For each parameter there is an expected variation in the values, due to the stochastic nature of parameter estimation. Timings indicate very fast estimation duration even for large datasets, given homogeneous point density.

Dataset/Parameter	min	max	avg	median	std	var
PCID5_Area_1						
ε (m)	0.035	0.038	0.036	0.036	0.001	5.26E-07
t_ε (s)	0.200	0.654	0.221	0.210	0.048	2.30E-03
τ (pts number)	35	42	37	38	1.637	2.68E+00
t_τ (s)	0.001	0.302	0.004	0.001	0.030	9.00E-04
PCID5_Area_2						
ε (m)	0.034	0.039	0.036	0.036	0.001	5.67E-07
t_ε (s)	0.224	0.319	0.247	0.242	0.018	3.00E-04
τ (pts number)	34	44	37	37	1.735	3.01E+00
t_τ (s)	0.001	0.002	0.001	0.001	0.000	4.09E-08
PCID5_Area_3						
ε (m)	0.034	0.038	0.036	0.036	0.001	4.50E-07
t_ε (s)	0.055	0.090	0.061	0.059	0.006	3.03E-05
τ (pts number)	34	42	38	38	1.524	2.32E+00
t_τ (s)	0.001	0.002	0.001	0.001	0.000	2.50E-08
PCID5_Area_4						
ε (m)	0.034	0.038	0.036	0.036	0.001	4.40E-07
t_ε (s)	0.198	0.268	0.222	0.224	0.016	2.00E-04
τ (pts number)	34	43	38	38	1.450	2.10E+00
t_τ (s)	0.001	0.002	0.001	0.001	0.000	3.56E-08
PCID5_Area_5						
ε (m)	0.035	0.039	0.036	0.036	0.001	4.59E-07
t_ε (s)	0.425	0.622	0.483	0.480	0.044	1.90E-03
τ (pts number)	34	43	38	38	1.552	2.41E+00
t_τ (s)	0.001	0.002	0.001	0.001	0.000	4.68E-08
PCID5_Area_6						
ε (m)	0.035	0.038	0.036	0.036	0.001	4.45E-07
t_ε (s)	0.177	0.332	0.199	0.197	0.020	4.00E-04
τ (pts number)	35	43	38	38	1.506	2.27E+00
t_τ (s)	0.001	0.002	0.001	0.001	0.000	2.16E-08
PCID6_REVO_INDOOR						
ε (m)	0.033	0.036	0.034	0.034	0.001	2.93E-07
t_ε (s)	0.129	0.171	0.143	0.141	0.009	7.27E-05
τ (pts number)	91	119	106	105	6.004	3.60E+01
t_τ (s)	0.004	0.011	0.005	0.005	0.001	5.69E-07

(continued on next page)

Table A.1 (*continued*)

Dataset/Parameter	min	max	avg	median	std	var
PCID6_REVO_OUTDOOR						
ε (m)	0.045	0.052	0.048	0.048	0.001	1.96E-06
t_ε (s)	0.089	0.136	0.107	0.106	0.009	8.86E-05
τ (pts number)	54	79	64	64	4.822	2.33E+01
t_τ (s)	0.002	0.014	0.003	0.003	0.001	1.51E-06
PCID8_NAAVIS_1						
ε (m)	0.019	0.021	0.020	0.020	0.000	1.50E-07
t_ε (s)	0.436	0.596	0.507	0.510	0.031	1.00E-03
τ (pts number)	14	19	16	16	0.863	7.45E-01
t_τ (s)	0.001	0.001	0.001	0.001	0.000	1.60E-08
PCID9_NAAVIS_2						
ε (m)	0.042	0.044	0.044	0.043	0.000	1.84E-07
t_ε (s)	0.022	0.032	0.025	0.025	0.002	4.12E-06
τ (pts number)	14	16	15	15	0.549	3.01E-01
t_τ (s)	0.001	0.001	0.001	0.001	0.000	1.20E-08
PCID10_RTWH_CHAIR						
ε (m)	0.007	0.007	0.007	0.007	0.000	2.98E-08
t_ε (s)	11.544	13.453	12.329	12.308	0.362	1.31E-01
τ (pts number)	26	35	30	30	1.824	3.33E+00
t_τ (s)	0.002	0.003	0.002	0.002	0.000	5.27E-08

References

[1] Q. Wang, M.K. Kim, Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018, Adv. Eng. Inform. 39 (2019) 306–319, https://doi.org/10.1016/j.aei.2019.02.007.

[2] X. Xiong, A. Adan, B. Akinci, D. Huber, Automatic creation of semantically rich 3D building models from laser scanner data, Autom. Constr. 31 (2013) 325–337, https://doi.org/10.1016/j.autcon.2012.10.006.

[3] R. Vanlande, C. Nicolle, C. Cruz, IFC and building lifecycle management, Autom. Constr. 18 (2008) 70–78, https://doi.org/10.1016/j.autcon.2008.05.001.

[4] R. Liu, V.K. Asari, 3D indoor scene reconstruction and change detection for robotic sensing and navigation, in: S.S. Agaian, S.A. Jassim (Eds.), SPIE - the International Society for Optical Engineering, International Society for Optics and Photonics, 2017, p. 102210D, https://doi.org/10.1117/12.2262831.

[5] F. Poux, R. Billen, Voxel-based 3D point cloud semantic segmentation: unsupervised geometric and relationship featuring vs deep learning methods, ISPRS Int. J. Geo Inf. 8 (2019) 213, https://doi.org/10.3390/ijgi8050213.

[6] R. Schnabel, R. Wahl, R. Klein, Efficient RANSAC for point cloud shape detection, Computer Graphics Forum. 26 (2007) 214–226, https://doi.org/10.1111/j.1467-8659.2007.01016.x.

[7] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, in: Conference on Neural Information Processing Systems (NIPS), Long Beach, United States, 2017. http://arxiv.org/abs/1706.02413 (accessed March 13, 2018).

[8] L. Landrieu, M. Simonovsky, Large-Scale Point Cloud Semantic Segmentation with Superpoint Graphs, in: Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, United States, 2018, pp. 4558–4567, https://doi.org/10.1109/CVPR.2018.00479.

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

[10] C. Choy, J. Gwak, S. Savarese, 4D spatio-temporal convnets: Minkowski convolutional neural networks, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019, pp. 3070–3079, https://doi.org/10.1109/CVPR.2019.00319.

[11] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, RandLA-Net: Efficient semantic segmentation of large-scale point clouds, in: ArXiv, 2019, pp. 11105–11114, https://doi.org/10.1109/CVPR42600.2020.01112.

[12] F. Matrone, A. Lingua, R. Pierdicca, E.S. Malinverni, M. Paolanti, E. Grilli, F. Remondino, A. Murtiyoso, T. Landes, A Benchmark for Large-scale Heritage Point Cloud Semantic Segmentation, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, International Society for Photogrammetry and Remote Sensing, 2020, pp. 1419–1426, https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1419-2020.

[13] A.F. Obrist, A. Flisch, J. Hofmann, Point cloud reconstruction with sub-pixel accuracy by slice-adaptive thresholding of X-ray computed tomography images, NDT and E International. 37 (2004) 373–380, https://doi.org/10.1016/j.ndteint.2003.11.002.

[14] K. Zhang, W. Bi, X. Zhang, X. Fu, K. Zhou, L. Zhu, A new Kmeans clustering algorithm for point cloud, international journal of hybrid information, Technology. 8 (2015) 157–170, https://doi.org/10.14257/ijhit.2015.8.9.16.

[15] J.M. Biosca, J.L. Lerma, Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods, ISPRS J. Photogramm. Remote Sens. 63 (2008) 84–98, https://doi.org/10.1016/j.isprsjprs.2007.07.010.

[16] T. Melzer, Non-parametric segmentation of ALS point clouds using mean shift, Journal of Applied Geodesy. 1 (2008) 159–170, https://doi.org/10.1515/jag.2007.018.

[17] J. Strom, A. Richardson, E. Olson, Graph-based segmentation for colored 3D laser point clouds, in: IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings, 2010, pp. 2131–2136, https://doi.org/10.1109/IROS.2010.5650459.

[18] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, A. Frenkel, On the Segmentation of 3D LIDAR Point Clouds, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, Shangai, 2011, pp. 1–8, https://doi.org/10.1109/ICRA.2011.5979818.

[19] E. Grilli, F. Menna, F. Remondino, A review of point clouds segmentation and classification algorithms, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 2017, pp. 339–344, https://doi.org/10.5194/isprs-archives-XLII-2-W3–339-2017.

[20] R. Rostami, F.S. Bashiri, B. Rostami, Z. Yu, A survey on data-driven 3D shape descriptors, Computer Graphics Forum. 00 (2018) 1–38, https://doi.org/10.1111/cgf.13536.

[21] Y. Xie, J. Tian, X.X. Zhu, A Review of Point Cloud Semantic Segmentation, ArXiv. Org., 2019. http://arxiv.org/abs/1908.08854.

[22] A. Kaiser, J.A. Ybanez Zepeda, T. Boubekeur, A survey of simple geometric primitives detection methods for captured 3D data, Computer Graphics Forum. 38 (2019) 167–196, https://doi.org/10.1111/cgf.13451.

[23] J. Deschaud, A fast and accurate plane detection algorithm for large noisy point clouds using filtered normals and voxel growing, in: Symposium A Quarterly Journal In Modern Foreign Literatures, 2010, in: http://campwww.informatik.tu-muenchen.de/3DPVT2010/data/media/e-proceeding/papers/paper111.pdf.

[24] J. Xiao, J. Zhang, B. Adler, H. Zhang, J. Zhang, Three-dimensional point cloud plane segmentation in both structured and unstructured environments, Robot. Auton. Syst. 61 (2013) 1641–1652, https://doi.org/10.1016/j.robot.2013.07.001.

[25] A.V. Vo, L. Truong-Hong, D.F. Laefer, M. Bertolotto, Octree-based region growing for point cloud segmentation, ISPRS J. Photogramm. Remote Sens. 104 (2015) 88–100, https://doi.org/10.1016/j.isprsjprs.2015.01.011.

[26] Z. Dong, B. Yang, P. Hu, S. Scherer, An efficient global energy optimization approach for robust 3D plane segmentation of point clouds, ISPRS International Journal of Photogrammetry and Remote Sensing. 137 (2018) 112–133, https://doi.org/10.1016/j.isprsjprs.2018.01.013.

[27] T. Rabbani, F..A. Van den Heuvel, G. Vosselman, Segmentation of Point Clouds Using Smoothness Constraint, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, Dresden, 2006, pp. 248–253, in: http://www.isprs.org/proceedings/XXXVI/part5/paper/RABB_639.pdf (accessed May 28, 2013).

[28] A. Nurunnabi, D. Belton, G. West, Robust Segmentation in Laser Scanning 3D Point Cloud Data, in: International Conference on Digital Image Computing Techniques and Applications, IEEE, Fremantle, WA, 2012, pp. 1–8, https://doi.org/10.1109/DICTA.2012.6411672.

[29] D. Tóvári, N. Pfeifer, Segmentation based robust interpolation - a new approach to laser data filtering, in: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives 36, 2005, pp. 79–84.

[30] F. Poux, R. Neuville, G.-A. Nys, R. Billen, 3D point cloud semantic modelling: integrated framework for indoor spaces and furniture, Remote Sens. 10 (2018) 1412, https://doi.org/10.3390/rs10091412.

[31] P. Hough, METHOD AND MEANS FOR RECOGNIZING COMPLEX PATTERNS, US3069654A. http://www.google.com/patents/US3069654?printsec=description#v=onepage&q&f=false, 1960 (accessed April 19, 2021).

[32] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recogn. 13 (1981) 111–122, https://doi.org/10.1016/0031-3203(81)90009-1.

[33] F.A. Limberger, M.M. Oliveira, Real-time detection of planar regions in unorganized point clouds, Pattern Recogn. 48 (2015) 2043–2053, https://doi.org/10.1016/j.patcog.2014.12.020.
[34] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, The 3D Hough Transform for Plane Detection in Point Clouds: A Review and a New Accumulator Design, 3D Research vol. 02, 2011, pp. 1–13, https://doi.org/10.1007/3DRes.02(2011)3.
[35] M. Camurri, R. Vezzani, R. Cucchiara, 3D Hough transform for sphere recognition on point clouds: a systematic study and a new method proposal, Mach. Vis. Appl. 25 (2014) 1877–1891, https://doi.org/10.1007/s00138-014-0640-3.
[36] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (1981) 381–395, https://doi.org/10.1145/358669.358692.
[37] S. Choi, T. Kim, W. Yu, Performance evaluation of RANSAC family, British Machine Vision Conference, BMVC 2009 - Proceedings 24, 2009, pp. 271–300, https://doi.org/10.5244/C.23.81.
[38] H.L. Nguyen, D. Belton, P. Helmholz, Planar surface detection for sparse and heterogeneous mobile laser scanning point clouds, ISPRS J. Photogramm. Remote Sens. 151 (2019) 141–161, https://doi.org/10.1016/j.isprsjprs.2019.03.006.
[39] B. Xu, W. Jiang, J. Shan, J. Zhang, L. Li, Investigation on the weighted RANSAC approaches for building roof plane segmentation from LiDAR point clouds, Remote Sens. 8 (2016) 5, https://doi.org/10.3390/rs8010005.
[40] V. Sanchez, A. Zakhor, Planar 3D modeling of building interiors from point cloud data, in: International Conference on Image Processing (ICIP), IEEE, 2012, pp. 1777–1780, https://doi.org/10.1109/ICIP.2012.6467225.
[41] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, L. Tang, An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells, Remote Sens. 9 (2017) 433, https://doi.org/10.3390/rs9050433.
[42] A. Boulch, R. Marlet, Fast and robust normal estimation for point clouds with sharp features, Eurographics Symposium on Geometry Processing. 31 (2012) 1765–1774, https://doi.org/10.1111/j.1467-8659.2012.03181. x.
[43] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D Semantic Parsing of Large-Scale Indoor Spaces, in: Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, United States, 2016, pp. 1534–1543, https://doi.org/10.1109/CVPR.2016.170.
[44] V. Lehtola, H. Kaartinen, A. Nüchter, R. Kaijaluoto, A. Kukko, P. Litkey, E. Honkavaara, T. Rosnell, M. Vaaja, J.-P. Virtanen, M. Kurkela, A. El Issaoui, L. Zhu, A. Jaakkola, J. Hyyppä, Comparison of the selected state-of-the-art 3D indoor scanning and point cloud generation methods, Remote Sens. 9 (2017) 796, https://doi.org/10.3390/rs9080796.
[45] A. Kharroubi, R. Hajji, R. Billen, F. Poux, Classification and integration of massive 3d points clouds in a virtual reality (VR) environment, international archives of the photogrammetry, remote sensing and spatial, Inf. Sci. 42 (2019) 165–171, https://doi.org/10.5194/isprs-archives-XLII-2-W17-165-2019.
[46] M. Bassier, M. Vergauwen, F. Poux, Point cloud vs. mesh features for building interior classification, Remote Sensing 12 (2020) 2224, https://doi.org/10.3390/rs12142224.
[47] E. Grilli, F. Poux, F. Remondino, Unsupervised object-based clustering in support of supervised point-based 3d point cloud classification, Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. XLIII-B2-2 (2021) 471–478, https://doi.org/10.5194/isprs-archives-xliii-b2-2021-471-2021.
[48] Y. Bengio, Y. Lecun, G. Hinton, Deep learning for AI, Commun. ACM 64 (2021) 58–65, https://doi.org/10.1145/3448250.
[49] F. Poux, J.J. Ponciano, Self-learning ontology for instance segmentation of 3d indoor point cloud, in: ISPRS (Ed.), International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Copernicus Publications, Nice, 2020, pp. 309–316, https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-309-2020.