Pogo Sampling: How to best preserve information during manual point cloud labelling

Felix Eickeler, Florian Noichl

Technische Universität München, Lehrstuhl für Computergestützte Modellierung und Simulation

Abstract

A primary concern when creating efficient lifecycle management solutions is digitising the built environment. One of the most important aspects of digitisation is translating the geometry, which can be extracted using laser scanners or photogrammetric surveys, to a semantic model. While pure surface descriptions generated by the sensors provide great value during manual inspection, further processing of the surfaces is needed to obtain a semantically rich model. The foremost task is the separation and classification of objects, which can be achieved either by classical machine vision or deep learning tools such as PointNet[5], or DGCNN[14]. However, similar to other deep learning methods, the training process is the key to model adaptation and defines the classifier's quality. The data set is generally prepared as follows: (1) a survey is conducted which describes the geometry of a build environment (2) The surface, often represented as a point cloud, is manually separated, partitioned, (3) and presented to the workforce for manual object annotation. During this process, the data is down-sampled to enable smooth operation. This down-sampling alienates the dataset from the original and the actual training, where a different internal down-sampling is applied by all available networks. This paper investigates the effect of down- and up-sampling on the annotation process and presents a guideline for algorithm selection.

Keywords: Point clouds, labelling, ground truth, down-sampling

1 Motivation

In the current strive for more automatisation in the AEC Industry, seamless monitoring and modelling are keys to enabling advanced control schemes and analyses. Due to the uniqueness of building environments, this task is very diverse and, therefore a prime candidate to be supported by Artificial Intelligence (AI). Base data as known as ground truth, needs to be collected for training and validation. This also holds true for deterministic approaches, where implementation itself is in-depended, but the testing and quantification of the achievable precision require a known quantity for comparison.
As a result, a significant amount of work is invested into manual data preparation, which is time-consuming, error-prone, but crucial to the method's success. For the data-hungry deep learning methods, techniques like unsupervised learning, transfer learning, active learning can minimise the amount of data needed. In the context of point clouds, this problem is prevalent as the data, unlike images or video frames, cannot be separated into non-interdependent entities without further reasoning or information loss. Furthermore, point clouds have a high information density, making the annotation process cumbersome, expensive, and turning data distribution among the workforce into a challenge.
However, as demand for annotated data is rapidly growing due to the increased interest of new industries and sectors previously not involved with AI application, the need for fast annotation and a reasonably priced workforce is immanent. Facing this requirement, point cloud data is usually down-sampled, i.e. the number of points is reduced. After annotation, these points can be merged into the main dataset. The whole process can be broken down into three major steps (see Figure 1): down-sampling, manual point annotation, and u-sampling. Since the annotated data has a lower resolution, some estimations must be made while transferring the annotation to the original cloud or the subset remaining with lower geometric detailing. Thus, with every point removed during sub-sampling, information is lost.

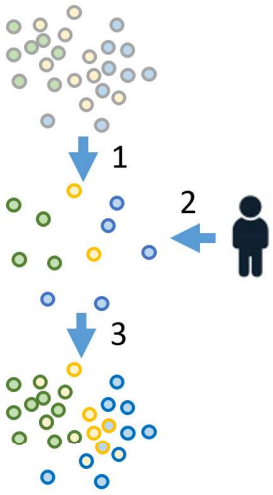

Figure 1: Overview 3 step process:
1 - down-sampling
2 - manual annotation
3 - up-sampling

In this contribution, we want to address this fact by analysing the different steps to minimise the immanent information loss. We use

a simplified model to investigate the error introduced by the interaction between the different down- and up-sampling algorithms (depicted by points with mismatching colours in 1). For this, we study various well-established methods and evaluate results for combinations.

2 State of the Art

Since the introduction of PointNet, a deep neural network for point cloud segmentation and classification, neural networks have been adopted for point cloud processing. Similar to any deep neural network, large-scale, annotated datasets are required to improve stability and generalisation. Typically this ground truth data is split into three parts: training, testing, and validation.
In the general field of Computer Vision, there are some widely popular examples of publicly available datasets like MNIST [12], and ImageNet [6] which help while developing and testing a task optimised model. For 3D application, the primary interest was by the automotive industry, which efforts in autonomous driving heavily on annotated point clouds of outdoor environments. A widely known example from this field is the Semantic KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) set [3]. For the AEC Industry, the problem is datasets showing complete indoor reconstruction, or the lack thereof. Besides the widely spread S3DIS (Standford 3D Indoor Dataset Stanford 3D Indoor Scene Dataset), an RGB-D dataset with 13 instance-wise annotated classes is available [1]. While this dataset is going in the right direction for the detection of office buildings, there is an increased interest in the application of deep learning techniques in industrial environments. However, unlike the office use case, no comprehensive, publicly and fully labelled datasets have been published[8].
To compensate for the lack of ground truth, especially when designing new use-cases or specialising detection tasks, simulation tools can be used as an alternative source of data. Aside from the tools that were made available, there are also numerous datasets based on simulation. An example would be SynthCity - a comprehensive urban environment [9]. While synthetic data alone might not be entirely sufficient due to a lack of realism, it offers the significant opportunity that labels can be included determined without any manual effort, therefore preventing expenses and manual errors in the process. This leads to a trade-off between diversity, environmental influences, noise and annotation quality and cost. Different approaches can be taken to create a more realistic simulation, one is to take the properties of the recording devices into account. For realistic simulations of laser scanners open-source tools such as Helios [2], or Blensor [10] allow to perform virtual laser scans using 3D object geometries, such as 3D building models. This introduces noise, shadowing, and a distance to resolution relationship, resulting in increased realism.
To annotate the raw data, users now have various tools at their disposal. For many applications in the context of point clouds, a popular option is the open-source tool of CloudCompare[1], which offers a range of ready-to-use implementations. Users can segment regions by drawing 2D shapes or boxes on free select-able planes. With this approach, two dimensions can always be selected at once. If labelling is more extensive, cloud-based applications such as AWS SageMaker GroundTruth[2] offer commercial solutions for the labelling of point cloud data. Depending on the task at hand, there are different options to perform data annotation or 'labelling' on point clouds.
While for object recognition tasks, the most common way of labelling is bounding boxes, especially towards semantic segmentation. Tools are required to depict more complex shapes. Common solutions are to draw polygons to edge class instances or to use brush-like tooltips with adaptive sizes to 'paint' clusters of points according to their semantics. For all mentioned methods, it is necessary to frequently change the perspective to avoid mislabelling and ensure the correct boundaries of each region of interest.

3 Methodology

The experiment tries to model the typical workflow of 3D annotation (outlined in Figure 2). The main goal is to select the most promising sampling method paring for the annotation process and quantify the error introduced by down- and up-sampling. For this, the annotation process is separated into three main components: down-sampling, classification of points, and up-sampling.

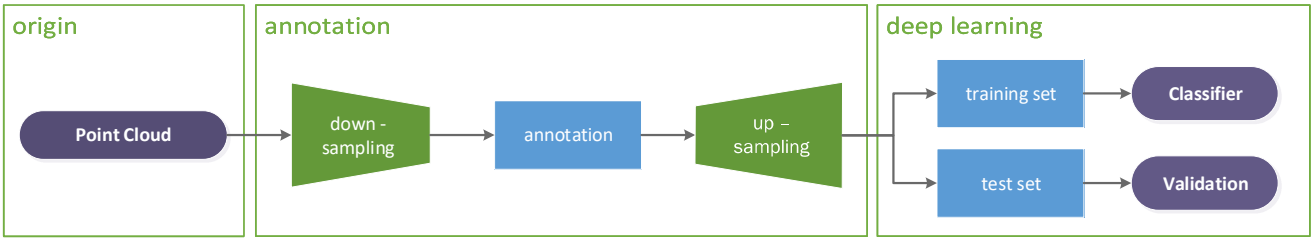

Figure 2: Generalised process of labelling 3D point clouds.

[1] http://www.cloudcompare.org/
[2] https://aws.amazon.com/sagemaker/groundtruth/

Before diving into the details of each step, the concept will be introduced: The first component of point cloud down-sampling, the following algorithms are investigated: random sampling, space sampling (a.k.a equidistant sampling), voxel grid and adaptive bilateral filtering. Each method has its own set of parameters. Therefore the relative number of points is used to normalise the filter's effect and make it a 'fair' comparison. The second component is the virtual labelling process which simulates the classification process conducted during manual labelling or space partitioning algorithms. Since it is impossible to model true human behaviour, we simplified the process to an optimisation that provides comprehensible and stable results. In contrast to the first component, the third component up-samples the point cloud to the original format. We use different techniques for the up-sampling processes again, and based on the relative number of points, the increase in resolution is normalised between the different techniques.
Since the input consists of a selected set of simulated, segmented, high-density point clouds, the result of this process can be directly compared to the original input. The experiment is set up, so the resulting segmentation will always include all points of the object (recall 100%).

3.1 Synthetic ground truth

Amongst others, we use data created using a complete toolchain as described in [13], including the Helios simulation toolkit, as introduced in 2. This method provides the correct ground truth, as the correct label is included on a point-by-point basis by the object semantics defined in the model before. This is important as we also want to evaluate a theoretical measure for the labelling expense.

3.2 Down-Sampling

Different down-sampling algorithms are used to reduce the number of points in the point cloud. While many different approaches for down-sampling exist, we focus on three contenders because of their wide use and availability as well as our personal experience (see Table 1). The algorithms can be divided in three categories: random reduction, linear space aware filters, and resolution sampling. The space-aware filters used for this study show emphasis on different properties. The equidistant sampling method, a filter that removes points until all points have a similar spacing, retains an excellent level of quality. However, while the most affect the high-density regions are parts that are closer to the capturing device, some regions of high object diversity often have structured surfaces, and essential points may be removed [7]. The voxel grid filter retains a centre point of a voxel if a certain threshold of points is reached. While most properties are similar to the equidistant spacing, the processing speed and the encoding simplification make it one of the go-to filters in the construction industry. In our implementation of up-and down-sampling, we make use of functionalities provided in the Open3D library [15].

3.3 Virtual labelling

The simplified annotation process assumes that points classification is conducted in 3 dimensions, where one dimension is permanently reduced by a projection of the point cloud (e.g. a side view will not show depth). The modelled process presents itself as a series of these 2-dimensional projections, in which stencils are applied to annotate the point cloud. For each stencil, the first and second projection planes can be selected freely, while the third projection will be linearly independent. While multiple stencil types are possible, we have chosen a projected cuboid (e.g. a projected bounding box) for this process. This process assumes that the selected cuboid will be chosen so that the number of needed stencils is minimal, e.g. the introduced error is minimised. In simpler words: "try to crop the object with k-boxes. Try to include as few wrong points as possible".

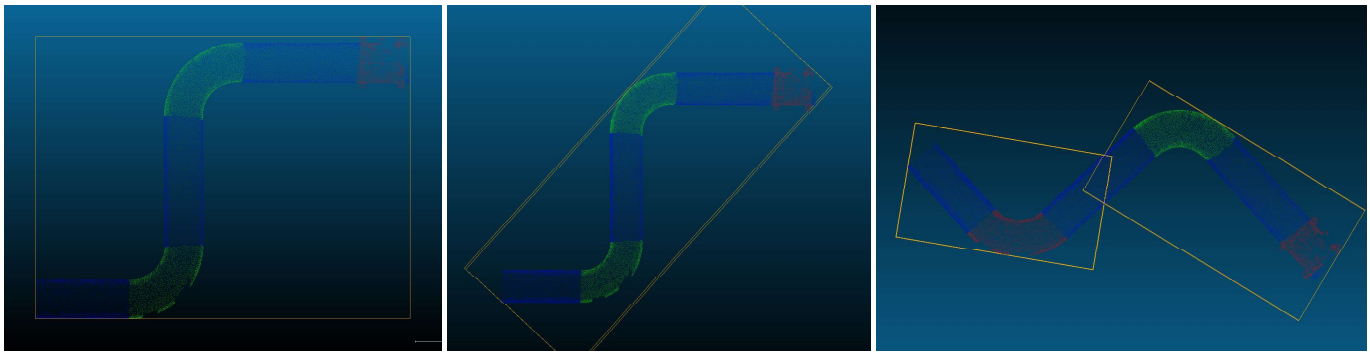

Figure 3: Process of annotation. **left -** original point cloud; **middle -** aligned bounding box; **right -** cut & aligned bounding box (first iteration complete). Since the pipe is non-twisted, their will be no benefit in slicing the point cloud along any other axis. Thus these branches will not be chosen and all further slicing will be in the shown plane.

Based on Huebner, Ruthotto, and Kragic [11] the following algorithms has been deducted (see Equation 3.3). A single object is selected from the simulated but down-sampled point cloud. This isolated object is then iteratively decomposed with bounding boxes:

The process starts by fitting the global (level zero, $k = 0$) bounding box and initialising a decision tree. The decision tree is necessary since our hypothesis selects the *best* stencil, which can only be evaluated in retrospective by comparing all projections of the current level. The object's points are projected to the three planes defined by the perimeter of the bounding box (denoted as A, B, C). Each projection of the three projections is rasterised by a fixed square grid, leading to a fixed resolution in each direction of the plane. For every plane, the optimal cut is calculated along the remaining two dimensions (denoted as indices a, b, c), resulting in 6 possible cuts and leading two six new possible bounding box configurations. Similar to the original paper, we use the area of the projection to select the best cut. This cut is translated to 3D, and two all-new bounding boxes are fitted (C_1, C_2. If the volume gain of this cut is sufficient, the box is accepted and entered into the decision tree. If a bounding box pair is rejected, the current level is marked as final. The gain is defined as:

$$\theta = \frac{V(C_1) + V(C_2) + V_{remaining}}{V_{k-1}} \tag{1}$$

where: $V_{remaining}$ describes all other uncut boxes and V_{k-1} describes the volume of the parent level.

If a box is not marked final, the process is repeated iteratively until a suitable amount of box-levels is reached. This metric can be either defined by a minimum volume gain or by the number of stencils that should be applied by limiting the number of cuts (k). The whole decomposition returns a series of freely defined bounding boxes that optimally enclose the object points.

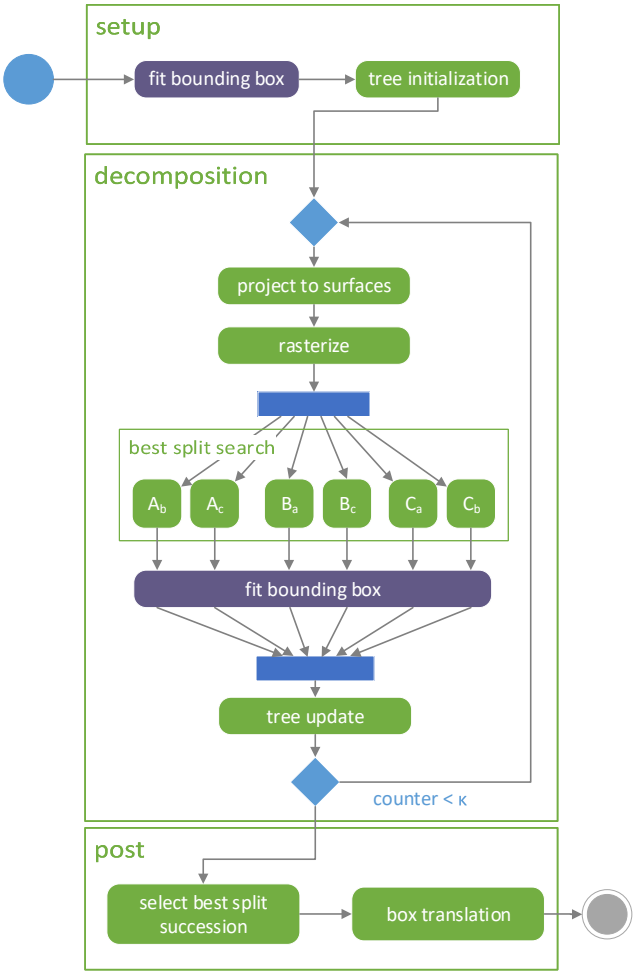

Figure 4: *Activity Diagram* of the algorithm described in subsection 3.3. The algorithm consists of a setup, κ-decomposition steps and post evaluation and bounding box translation steps.

Calculation Effort: The overall cost of calculation is relatively low, but the tree might grow with 6^k where k is the level of cuts that are targeted for the decomposition. Since we try to model primarily manual labour, the number of decomposition steps can be

considered low. Additionally, two possible bounding boxes are of interest: the minimum volume bounding box (mvbb) and the primary component aligned bounding box (pca). While the first one will show better results, the problem is that a minimum volumetric bounding box is $\mathcal{NP}$-hard. We found that the mvbb fit takes around 1-2 seconds per level (k) for objects around or smaller than 100k points. The used mvbb algorithm was provided by CGAL [4].

3.4 Up-Sampling

The last generative process in our experiment is the up-sampling process. Up-sampling needs to estimate further dimensions (in our case points) from an existing low-resolution object. In our case is the geometry of the original known, as it is retained, and the up-sampling process only consists of one point property. We therefore have to provide $n_{new} = (n_{origal} - n_{retained})$ point properties. One important difference, and the main part of our investigation, is how the different up-sampling filters handle the two different preconditions: Were the points retained during down-sampling (random and equidistant sampling) or are points replaced (voxel grid, partially adaptive compression). The investigated techniques are all linear and evaluate the unknown points by evaluating the distance to:

1. nearest neighbour – Property defined by the nearest neighbour
2. nearest x-neighbours (NN-)voting – The majority of the x-nearest neighbours defines the property
3. next x-voxels – The majority of the x-nearest voxels neighbours defines the property
4. next x-stencils – The majority of the x-nearest stencils defines the property

3.5 Experiments & Parameters

The investigation includes multiple steps that influence the overall result. Since we use virtual labelling, we cannot compare the result to the semantics of the point cloud as we would include the quality of the virtual labelling process- which cannot be isolated from the sampling methods. We therefore conduct four different experiments isolating the influence of each step (see Figure 5).

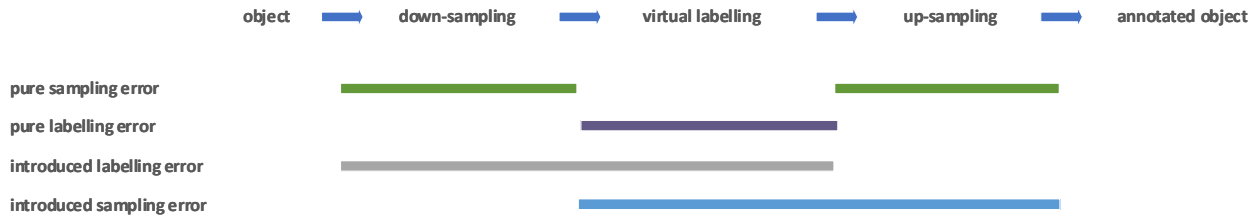

Figure 5: Description of the metrics used. Each metric spans a specific part of the process.

Pure Sampling Error - The pure sampling error (PSE) describes the information loss during down-sampling and up-sampling. It is the delta of the lost information during down-sampling and reconstructable information due to the up-sampling process. It describes the base error that sampling with perfect labelling would achieve.

$$PSE = \frac{P_{down+up}}{P_{original}} \tag{2}$$

Pure Labelling Error - The pure labelling error (PLE) describes the error that is introduced by the virtual labelling techniques. Boxes might span wrongly affiliated spaces since only the object itself is considered during the virtual labelling process (e.g. points of a tabletop might be included with the object of interest being located on the table). It describes the ability of the boxes to follow the object contours. It is highly dependent on the number of boxes (limited by the number of splits k). It is evaluated using the full, unfiltered point cloud.

$$PLE = \frac{P_{boxes \cap original}}{P_{original}} \tag{3}$$

Introduced Labelling Error - The introduced labelling error (ILE) describes the additional labelling error that is introduced in the down-sampling process. It compares the virtually labelled boxes of the original with the down-sampled version. Out of simplicity, we compare the volume of each box (e.g. the selected space). The ILE measures how well the labelling process can be performed on down-sampled point clouds.

$$ILE = \sum_{0}^{n} \frac{|V_i(C_{original}) - V_i(C_{down})|}{V_i(C_{original})} \tag{4}$$

Introduced Sampling Error - The introduced sampling error (ISE) describes the information loss after the up-sampling process. Similar to the PLE, ISE is a metric for the capabilities of the up-sampling process to reverse the effects of down-sampling. However, the ISE includes the annotation process (in contrast to PLE). We use the original boxes for object segmentation and then analyse the difference between the original and the sampled semantics.

$$ISE = \frac{\Delta P_{(boxes \cap down)+up}}{P_{boxes \cap original}} \tag{5}$$

Parameters

Since our toolchain is fully automated, parameter combinations can be tested with little effort. Table 1 shows the selected parameters and the ranges that were applied for this experiment. The parameter set is divided into two sets: actively controlled parameters and passively controlled parameters. The actively controlled parameters are used during the model run by the algorithm and can be varied freely during the study. They include compression ratio ν, number of boxes κ and the false discovery rate FDR. The passive parameter defines the setup for this experiment. They can be influenced by using different geometric models, point cloud simulation tools or settings. They are can also not be controlled directly and are not independent (e.g. as we cannot determine the density without simulation, with simulation parameters also changing the density of the point cloud). The passive parameters are part of the test set and remain fixed during the parameter study. In addition to the shown parameters, all down- and up-sampling algorithms have additional settings, which remain fixed.

Table 1: Input parameters describing the possible variation of the experiment:

Input Parameters			
name	lower bound	upper bound	description
sampling algorithms	random sampling space sampling	nearest neighbour NN-voting	Pairing of up- and down-sample algorithm.
point segments	-	-	number and shape of segments provided by the test data set.
point density (ϱ)	0.001	0.005	average density of the test data set measured by NN-5 (5 nearest neighbours) by the test data set. Values roughly mimic real world densities.
compression ratio (ν)	10%	20%	retaining factor for the down-sampling and the up-sampling process $\frac{down\ samples}{original}$.
number of boxes (κ)	1	8	applied boxes for decomposition (annotation).
false discovery rate (FDR)	95%	50%	measure for the adaption of the stencil annotation concept to the real segment. It is calculate by $\frac{false-positives}{positive-discoveries}$.

4 Results

For an illustration of the interaction between the different components and the evaluation method, the Stanford Bunny is used (section 4). We reduced down- and up-sampling techniques to two variants each to keep the extent of the case study and results manageable for the scope of this paper. As introduced, we investigate down-sampling with compression ratios of ν with 0.1 and 0.2. In random sampling, this can be achieved directly by defining the target number of remaining points. In space sampling, however, the sampling parameter of minimum distance that leads to the desired compression ratio needs to be designated by trial. For the case of the Stanford bunny, this could be achieved by a minimum distance of 0.137 ($\nu = 0.2$) and 0.194 ($\nu = 0.1$). The results of our evaluation metrics are presented in Tables 2, 3, 4 and 5. For the sake of compactness, the results are represented with their mean and respective standard deviation over all objects.

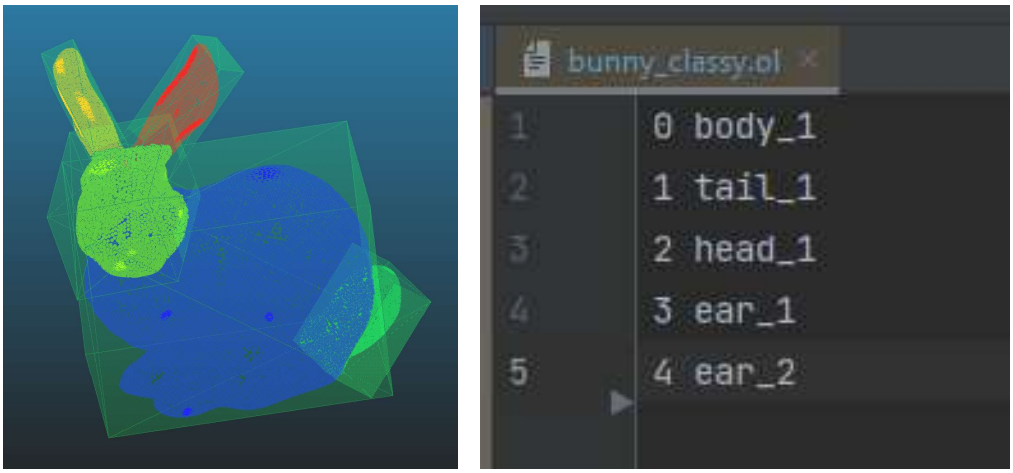

Figure 6: **left** – Stanford bunny with segmented objects ($\kappa = 0$). The bounding boxes shown are the minimal volume oriented bounding boxes. **right** – Associated classes.

Table 2: Evaluation of the PSE

down-sampling algorithm	ν	*up-sampling algorithm*	$PSE\,[\%]$
random	0.1	nearest neighbour	1.44
		NN-voting	1.89
	0.2	nearest neighbour	1.03
		NN-voting	1.31
minimum distance	0.1	nearest neighbour	1.63
		NN-voting	1.51
	0.2	nearest neighbour	1.18
		NN-voting	1.16

Table 3: Evaluation of the PLE

κ	*accepted boxes*	*volume* $[\%]$	*volume* $[SD\,\%]$	$PLE[\%]$	$PLE\,[SD\,\%]$
0	1	100	0	13.66	10.95
1	2	87.29	5.49	9.48	6.78
2	4	77.11	7.53	8.60	6.67
3	6.8	64.47	6.47	5.96	4.20
4	10.2	52.68	10.41	-	-

Table 4: Evaluation of the ILE

down-sampling algorithm	ν	κ	$ILE\,[\%]$	$ILE\,[SD\,\%]$
random	0.1	0	3.93	1.93
		1	6.44	3.22
		2	8.28	3.80
		3	11.48	4.75
	0.2	0	3.52	2.04
		1	3.60	1.23
		2	7.76	5.83
		3	6.55	3.18
minimum distance	0.1	0	5.02	2.46
		1	6.08	2.74
		2	4.86	2.32
		3	6.90	2.87
	0.2	0	7.72	3.35
		1	12.60	5.78
		2	12.53	6.16
		3	13.07	6.53

Table 5: Evaluation of the ISE

down-sampling algorithm	ν	*up-sampling algorithm*	$ISE\,[\%]$	$ISE\,[SD\,\%]$
random	0.1	nearest neighbour	1.28	2.27
		NN-voting	1.06	1.56
	0.2	nearest neighbour	1.17	1.99
		NN-voting	1.03	1.84
minimum distance	0.1	nearest neighbour	1.44	1.99
		NN-voting	1.44	1.99
	0.2	nearest neighbour	1.24	1.82
		NN-voting	1.33	1.82

4.1 Error Evaluation

The experiment results show that the annotation quality is mainly governed by the influence of the bounding box shape. This influence is visible when comparing the influence of the PLE with the previously defined metrics. However, while the shape segmentation with only bounding boxes is not optimal, the increase of accepted bounding boxes (bounding boxes that provide a particular improvement) diminishes after $\kappa = 4$. Comparing the average accepted boxes 6.8 with the theoretical maximal number of 16 indicated that the gain of annotation quality is low compared to the increasing labelling cost. Additionally, explicit object types or the object's class has a strong impact on the labelling quality. This impact can be observed by the high standard deviation of the PLE. For example, if the object is a wall, an aligned bounding box represents most points.

The PSE (Table 2) shows that even with a relatively high number of points, e.g. a compression ratio of 0.2, the minimum quality loss is over 1%. While 1% does not sound like much, gaining these gains in precision or recall through a neural network is a much greater effort. The best pure sampling combination is the $\varphi = 0.2$, nearest neighbour, with random sampling. In this, NN-voting with five neighbours leads to an error reduction for randomly down-sampled clouds. In contrast, the clouds down-sampled using space sampling show worse properties.

We evaluate the ILE (Table 4), the overall down-sampling, labelling and up-sampling scenarios. The ILE error is rising with a growing number of splits in the labelling process. While the error is elevated for *random sampling* with $\nu = 0.1$ and $\nu = 0.2$, the opposite of this behaviour is visible in the resulting error for the *minimum distance sampling*.

Untouched by the results of the labelling process, the ISE indicates similar behaviour. Interestingly the NN-voting seems to work very well with random sampling. This combination provides a good shape estimation of the bounding box and the lowest deviation. As a benefit, it seems to be very reluctant to the compression ratio.

Overall the influence of the down-sampling methods is relatively low. The small influence could be caused by the uniform point distribution, which is the case for the used dataset.

5 Discussion and Outlook

In this work, we present a concept to evaluate the influence of down- and up-sampling on the labelling process. We defined several key indicators to quantify the information loss over different process steps: PSE, PLE, ILE and the ISE. With a small proof-of-concept study, we indicated that this setup is capable of in-depth analysis and help to determine the most valued combination in different scenarios. Based on the use-case of new annotation or refurbishing existing labels, the processing steps can be selected with greater care leading to accuracy gains shy of 2%. While it does not seem much at first glance, such gains are costly in well-developed AI systems. Achieving this by just changing the sampling techniques is a minor investment. Additionally to the annotation concept, the results are a strategic thrust to select the best technique.
In future steps, further datasets need to be investigated and sampling techniques added. Expanding the dataset includes using different data sources, such as laser scanners, structure-from-motion, RGB-D cameras. The introduced metrics, however, will be kept the same as they comprehensively cover the entire toolchain. Ideally, this analysis needs to be performed once, and future annotation can leverage these results by transfer learning or incorporating elements of a data set. Choosing the proper sampling methods to avoid aliasing can improve the predictions and prepare for future deep learning trends when densities may increase.
With this research, we complement the down-sampling research used in AI systems. Internally most AI-Systems down-sample the point for further processing. For example, PointNet++ uses farthest point sampling to deal with overly dense point clouds. The authors think it is essential to increase the quantification of the influence of the different sampling methods to select or modify competitive strategies. Indeed, in our option, fast and efficient down-sampling methods for large scenes will significantly impact future approaches in training and annotation in the near future.

References

[1] Iro Armeni et al. “3D semantic parsing of large-scale indoor spaces”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 4 (2016), pp. 1534–1543. ISSN: 10636919. DOI: 10.1109/CVPR.2016.170.

[2] S. Bechtold and B. Höfle. “{HELIOS}: A Multi-Purpose LiDAR Simulation Framework for Research, Planning and Training of Laser Scanning Operations with Airborne, Ground-Based Mobile and Stationary Platforms”. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* III-3 (June 2016). DOI: 10.5194/isprsannals-iii-3-161-2016.

[3] J. Behley et al. “SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences”. In: *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*. 2019.

[4] Chia-Tche Chang, Bastien Gorissen, and Samuel Melchior. “Fast Oriented Bounding Box Optimization on the Rotation Group SO(3,R)”. In: *ACM Trans. Graph.* 30.5 (Oct. 2011). ISSN: 0730-0301. DOI: 10.1145/2019627.2019641.

[5] R. Qi Charles et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 77–85. DOI: 10.1109/CVPR.2017.16.

[6] Jia Deng et al. “ImageNet: A large-scale hierarchical image database”. In: June (2010), pp. 248–255. DOI: 10.1109/cvpr.2009.5206848.

[7] F. Eickeler, A. Sanchez-Rodriguez, and A. Borrmann. “Adaptive feature-conserving compression for large scale point clouds”. In: *Advanced Engineering Informatics* 48 (2021), p. 101236. DOI: 10.1016/j.aei.2020.101236.

[8] Biao Gao et al. “Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods”. In: *IEEE Transactions on Intelligent Transportation Systems* (2021), pp. 1–19. ISSN: 1524-9050. DOI: 10.1109/tits.2021.3076844.

[9] David Griffiths and Jan Boehm. “SynthCity: A large scale synthetic point cloud”. In: (2019), pp. 1–6. arXiv: 1907.04758. URL: http://arxiv.org/abs/1907.04758.

[10] Michael Gschwandtner et al. “BlenSor: Blender sensor simulation toolbox”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 6939 LNCS.PART 2 (2011), pp. 199–208. ISSN: 03029743. DOI: 10.1007/978-3-642-24031-7_20.

[11] Kai Huebner, Steffen Ruthotto, and Danica Kragic. “Minimum volume bounding box decomposition for shape approximation in robot grasping”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2008), pp. 1628–1633. ISSN: 10504729. DOI: 10.1109/ROBOT.2008.4543434.

[12] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[13] Florian Noichl, Alexander Braun, and André Borrmann. “"BIM-to-Scan" for Scan-to-BIM: Generating Realistic Synthetic Ground Truth Point Clouds based on Industrial 3D Models”. In: *Proceedings - 2021 European Conference on Computing in Construction* (2021).

[14] Yue Wang et al. *Dynamic Graph CNN for Learning on Point Clouds*. 2019. arXiv: 1801.07829 [cs.CV].

[15] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. “Open3D: A Modern Library for 3D Data Processing”. In: *arXiv:1801.09847* (2018).