© 2022 Society of Photo-Optical Instrumentation Engineers.
One print or electronic copy may be made for personal use only. Systematic reproduction and distribution, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited.

Overview of 3D point cloud annotation and segmentation techniques for smart city applications

Dominik Merkle[a,b] and Alexander Reiterer[a,b]

[a]Albert-Ludwigs-Universität Freiburg, Department of Sustainable Systems Engineering - INATECH, Emmy-Noether-Straße 2, Freiburg im Breisgau, Germany
[b]Fraunhofer Institute for Physical Measurement Techniques IPM, Georges-Köhler-Allee 301, Freiburg im Breisgau, Germany

ABSTRACT

The automated segmentation of photogrammetric or LiDAR PC data offers the basis for further post-processing. One example is the use of enriched data for simulating wind, heat, and rain to increase the resilience of cities against stronger and more frequent weather events due to climate change. The semantic information can be used to distinguish tree species or surface types within simulations for more precise results. Apart from that, many other applications like autonomous driving or disaster management can benefit. However, the quality of the segmentation highly depends on the amount and quality of annotated training data and the applied segmentation method. This paper gives an overview and compares state-of-the-art PC segmentation methods including both 2D, 3D and combined approaches. Furthermore, current annotation tools and methods are evaluated regarding the accuracy, the number of classes, and time effort. Moreover, current strategies and methods to extend training data with augmented or synthetic data are listed and assessed. Besides, we present a concept for a specific segmentation task based on a trade-off of the previously listed methods. The task includes the segmentation and partial annotation of colourized PC data of two urban areas in Freiburg Germany, acquired from a driven and flying platform. Lastly, we present the required steps to extend the segmentation task from one city section to an overall city with diverse areas and to adapt to different sensor systems leading to different PC specifications. We conclude this paper by identifying challenges and required research in the field of PC segmentation and annotation.

Keywords: Semantic segmentation, 3D point cloud, Annotation, Labeling, LiDAR, Mobile mapping, Laser scanning, RGB

1. INTRODUCTION

Stopping the anthropogenic climate change is one of the major challenges of humanity. The risk of drought periods, high temperatures, storms, and flooding like in Ahrtal, Germany, in 2020 become more frequent. The direct consequences are injured people and deaths, destroyed infrastructure, houses, and property, and high heat loads. This leads to a burden on the health care system and immense costs for society and each individual. The dangers exist both in rural and urban environments. However, due to the high population density of cities, it is particularly important to make them resilient against those dangers. To do this, one must better understand the heat and wind loads and heavy rain events. For this purpose, physical simulations can be used. They suffer from long processing times which prevents fast and short-term analysis. Furthermore, multiple simulations of different scenarios are not possible. As part of our project I4C, machine learning (ML) is used to obtain faster simulation results. Apart from weather data and boundary conditions, geometrical information about the environment is required. Since structures in cities interact different for heat, wind, and heavy rain simulations, semantic information is required. As part of I4C the following classes must be distinguished: Unknown, Building, Wall, Conifer, Deciduous tree, Shrubs, Hedge, Grass, Other vegetation, Sealed surface, Cobbles, and Open surface.

Further author information: (Send correspondence to Dominik Merkle)
Dominik Merkle: E-mail: dominik.merkle@ipm.fraunhofer.de, Telephone: +49 (0)761 8857-145
Alexander Reiterer: Email: alexander.reiterer@ipm.fraunhofer.de, Telephone: +49 (0)761 8857-183

in: Erbertseder, T.; Chrysoulakis, N.; Zhang, Y.; Society of Photo-Optical Instrumentation Engineers – SPIE: Remote Sensing Technologies and Applications in Urban Environments VII. Proc. SPIE 12269, Paper 1226903 (2022); doi: 10.1117/12.2635845

Some classes are expected to be detectable only from the geometry information. Others are probably not distinguishable without RGB information. Especially the surfaces might also need the 2D image information since it requires a high resolution.

The test city is Freiburg. For the simulations, two climatically interesting areas are selected. The two areas of around 300 × 300 m are depicted in Figure 1. As can be seen, the vegetation and the type of houses are different. LiDAR and camera data of the two areas are acquired using a mobile mapping vehicle. This means that the perspective is from the street. Therefore, only parts of the environment are included as can be seen in an exemplary point cloud (PC) in Figure 2a. Roofs, courtyards, and the backside of e.g. houses are missing. It makes sense to use a different data source from a different perspective. Data acquired from an aircraft is shown in Figure 2b. Here, the street, courtyards, and roofs are included but vertical structures and surfaces such as house walls are missing. Merging both data gives a more complete representation of the environment as depicted in Figure 2c. Disadvantages are the combination of different data quality like point density, noise level, accuracy, intensity or RGB information, and double structures due to offsets in georeferencing. Moreover, some data may have artefacts as shown in Figure 3. This must be taken into account during annotation and also segmentation of the data. In this case, the airborne-based data can include uneven PCs of flat surfaces like streets. Moreover, objects like cars are not sharp but have smoothed z-coordinates. Due to resolution issues, trees might not be included or only partially. Only the shadows might be included as projected RGB information. A more detailed comparison of airborne and mobile mapping data is given by William et al.[1] The depicted airborne-based data is acquired in wintertime when there are no leaves on the trees. For the driven data, both a summer and winter drive are done to include both seasons.

2. DATA ANNOTATION

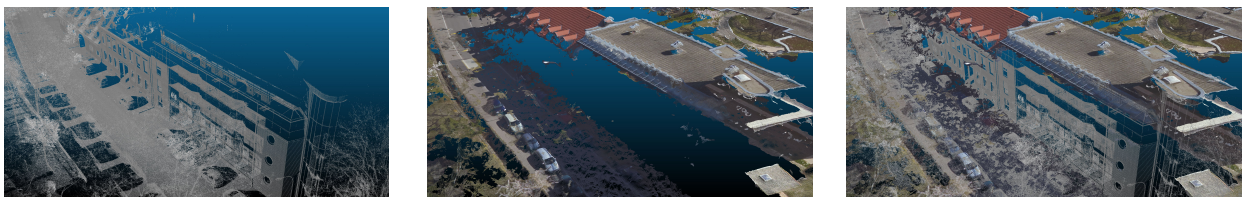
2.1 Related work

The backbone of a reliable PC segmentation is the quantity and quality of the training data. Both come at the expense of time effort. Therefore, the annotation should be efficient. One way of doing the labelling is annotating 2D image data and projecting the 3D data into the images. This is straight forward but this means



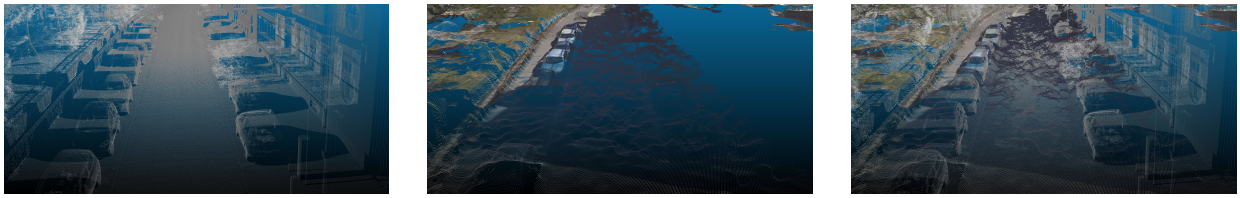
a) Test area 1.[2] b) Test area 2.[3]

Figure 1: Selected test areas.



a) PC from driven platform. b) PC from airplane. c) Merged PC.

Figure 2: Merging of PC data acquired from a driving and flying platform.



a) PC from driven platform. b) PC from airplane. c) Merged PC.

Figure 3: Detail view of merging of PC data acquired from a driving and flying platform.

that many images have to be annotated to cover the overall PC data. Moreover, due to occlusion, there might be false projections such as tree-labels on a house wall or shadowing effects leading to unlabeled points. Currently, the trend goes to annotating directly in 3D. The two major ways of cropping PCs are using bounding boxes or polygons in the current projected view in visualisation software. For the latter, the perspective has to be rotated usually at least once to cut from different views. Since this approach is time-consuming, advanced methods are studied. Wang et al.[4] present a one-click annotation method using sensor fusion and image-based detection algorithms to reduce time effort. Apart from the time factor, annotation in 3D is exhausting. Franzluebbers et al.[5] present a method of labeling using virtual reality. Bacim et al.[6] present a free-hand gesture interface. Guo et al.[7] list available data sets and networks for 3D shape classification, object detection and tracking, and segmentation.

The efficient annotation of PCs requires software. Yan et al. present a tool for PC annotation.[8] It is a semi-automatic approach using clustering in advance to allow annotating a cluster with one click. Another tool is given by Ibrahim et al.[9] called PC-Annotate. Moreover, they give an overview of further commercial and open-source annotation tools. Another tool is Amazon Sagemaker Ground Truth which is a function of Amazon SageMake.[10] CloudCompare[11] is a PC and mesh processing software. It can be also used for annotation using the segmentation tool. By defining scalar values or naming segmented PCs, the data can be prepared for training. Based on the annotation approach, both segmentation and instance segmentation are possible. For instance segmentation, either each instance must be annotated individually or post-processing based on clustering can be used. However, this only works well for separated structures. Trees that overlap each other are more difficult. Here, it helps to use a fitting function since a tree usually has only one trunk, though the green of the overlapping trees would have to be clustered based on RGB information.

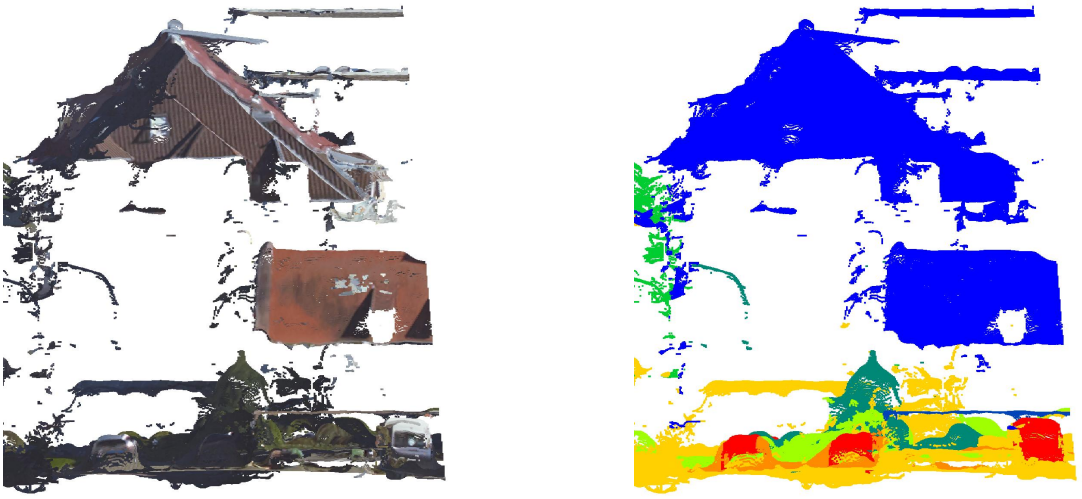
2.2 Strategy for I4C

For I4C, the available airborne data partially suffer from poor quality and artefacts. The correction of it can be either done directly manually or using filters or after annotation when e.g. streets can be easily removing from PC data. One approach is annotating sealed surfaces within the driven data first and removing double occurrences within the airborne data. Apart from the disadvantages of merging the data sources, annotating e.g. a house is faster within the merged data since the context for annotation is higher and both data can be annotated in one pass. Furthermore, it can be possible to annotate only the roofs of the airborne data and assign the building label to all points which are below. This can be done from the side perspective shown in Figure 4. In special cases, like a carport, the cobbles below might be assigned wrong as long as there is no enhanced algorithm that for example checks if there are openings. However, if these cases are rare, the speed gain of a large-scale annotation outweighs. Furthermore, they can be applied only partially for certain city regions where it is reasonable.

Figure 5 depicts the RGB and annotated PC from an angled view. Due to occlusion within the visualisation software, the sequence of annotation is important. Some tools have different view options, that allow only showing points in front of the virtual camera as possible in CloudCompare.[11] However, one would have to zoom very close to use the benefit which is not practicable in contrast to indoor annotation where it is a helpful feature. Since the ground in the selected areas is relatively flat, the classes divide into surfaces and 3D structures. The surfaces consist of Grass, Sealed surface, Cobbles, and Open surface. The 3D structures include Building, Wall, Conifer, Deciduous tree, Shrub, and Hedge. The class Unknown can be both. Cars are no class used

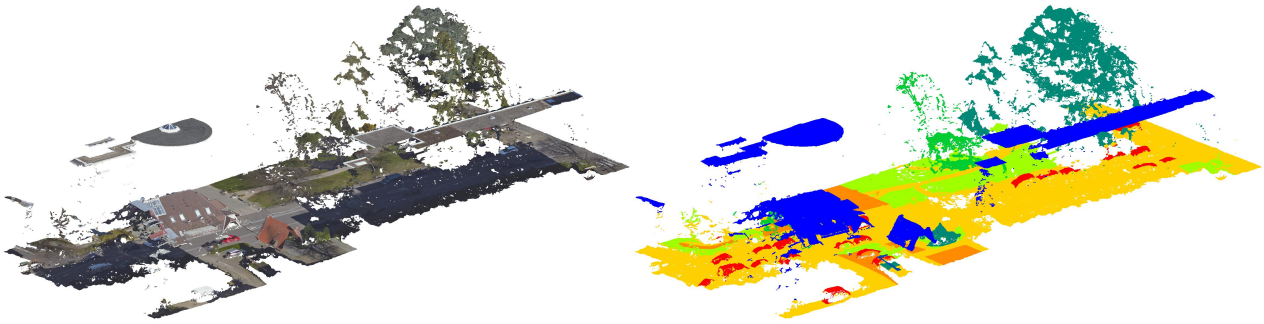
for simulation. However, its annotation is a result of segmenting the street, that belongs to Sealed surface. Regarding the sequence, it makes sense to first annotate the big and easy structures like Building and trees since they occlude most of the other classes. Cutting buildings generously from the top and subsequently correcting the interface to the ground surface or laterally adjacent objects is straightforward. The same applies to a single tree or a group of it. Here, the challenging part is the distinction between Conifer and Deciduous tree. If both the summer and winter data are available, only the super category tree has to be annotated. The type can be automatically derived by the difference of summer and winter data. This assumes that there is only a little or no snow covering the trees. For the Conifer, the difference should be small compared to the difference for the Deciduous tree. In case, only the winter data is available, the amount of green should be low for leave trees. If only the summer data is available, the 2D images might be required if the resolution is not sufficient to represent the leave and branch structure of the trees. The next challenge is the distinction between Wall, Shrub and Hedge. Vegetated walls are easily confused with hedges. In contrast to shrubs which are separately planted with a minimum distance, hedges are usually planted densely and build a linear or curved structure. In the future, the greening of vertical structures will be important for the climate in the cities. Here, the LiDAR data might help to check the penetrations. Otherwise, data from the two different seasons can help too since for greening mostly plants with leaves are used. A further challenge is to distinguish between Wall and a wall which is part of a building hence the definition must be clear before the annotation starts. One possible definition is, that a wall must not have contact with a building. Since this is often the case, it is better to define Wall as a structure that has no load-bearing function of a building and can be seen as an additional or independent structure. The efficient sequence of the 3D structures is dependent on the environment and the preference of the human annotator. Since monotonous and repetitive work can be exhausting for humans, it makes sense to define higher-level rules but allow an individual annotation approach based on the 3D environment. After the large 3D structures are finished, it makes sense to also remove big 3D structures which belong to the class Unknown and not leave it until the very end. Now, most surfaces and small 3D structures are left. Some surfaces are still covered by small 3D structures like streets mainly by cars, Cobbles and pavement by street lights, benches, trash cans, or sculptures, and Grass or Open surface by artwork or playgrounds. Removing these objects manually costs the largest share of time. Therefore, one should check if annotating these objects as an additional class adds value to the project in general or is a further goal to use synergies. As part of I4C, it makes sense to add Car as an additional class since with the use of the segmentation network, cars in newly acquired data of other environments can be removed and the gaps below the cars could be filled. For the manual annotation, either the street without the cars can be segmented or the cars can be segmented and the rest belongs to the street. The latter is safer since it consists of multiple steps. Otherwise, in case of an operating error, the segmentation of a big part of the street must be repeated. This risk depends on the size of the cropped data that is annotated and must be weighed by the annotator.

For the annotation, huge city data should be cropped to a minimum of 50 $m \times 50$ m and a maximum of a view hundred meters. This is a trade-off between enough context and breaking the big annotation task into smaller tasks that leads to a feeling of success, reduces errors, and facilitates the finding of mistakes and errors. Moreover, dependent on the PC resolution, both GPU and CPU resources for annotation can be reduced. For quality assurance, an overlap of neighbouring areas is helpful. Since even, if there is only one annotator, volatile errors can be detected or reduced. Systematic errors, however, can only be detected if complete areas or parts are annotated by at least two independent people. It saves a lot of time to start with quality assurance as soon as possible to redefine the segmentation guidelines and prevent mistakes for the overall data set. The cropping of the overall data can be based on an equidistant grid with constant cell size. This has the benefit, that the number of points is approximately equal. Depending on the environment and the velocity of the mobile platform, this can still vary strongly in individual crops. Moreover, the equidistant approach makes the assignment easy and allows fast post-processing methods based on neighbouring conditions to merge the instances of a split house. Furthermore, structures of buildings predicted as Unknown due to the small share in this specific crop can later be assigned to the adjacent class of the neighbouring crop. Another way of cropping is cutting the area manually using polygons from the top view. This can save time if e.g. a huge area is only Grass or if there is a huge Building over several equidistant cells but lies diagonally to north or the desired grid orientation. In the case of an equidistant grid, big classes could be annotated in big crops and smaller classes within smaller crops. However, this approach can be resource-intensive. A further automatic approach is cropping relative to



a) Side view of RGB PC. b) Side view of annotated PC.

Figure 4: Example of large-scale annotation of buildings cutting multiple roofs in the airborne data in one pass. The instances can be derived afterwards by clustering.



a) Angled view of RGB PC. b) Angled view of annotated PC.

Figure 5: Annotation of PC with many objects leading to visual occlusion while annotating.

the vehicle position, based on driven distance, or time during the mapping process. For the data acquired by driving, distance-based cropping is independent of driving speed. A time-based cutting would ensure that the number of points is rather homogeneously distributed. In the case of I4C, each street is driven at least once per direction. Therefore, it would lead to partial overlapping crops. To keep it simple, for I4C the equidistant grid cropping is chosen. To be mentioned here is that the cropping for annotation does not have to be the cropping for segmentation since it can be chosen arbitrarily different. This does not apply for the point density. When annotating sparse downsampled PCs, the segmentation can not directly be done in full resolution. However, in some cases, it makes sense to reduce the point density for annotation due to limited hardware resources. To transfer the annotation to the full resolution, an interpolation approach or nearest neighbor approach can be used. The same approach can be applied to transfer annotations from summer data to winter data or vice versa. This can reduce the effort almost by half. Due to the the dynamic environment, structures and surfaces may change in between, cars park differently, the driven trajectory differs, or even the sensor could change. The latter is not recommendable. In this case the data of the better sensor should be adapted to the poorer data quality of the other sensor.

3. DATA SEGMENTATION

3.1 Related work

Once the PC data is acquired and annotated, the data must be processed to obtain semantic information. For some tasks, clustering, object detection, filtering, geometric processing, or other approaches are sufficient. For other tasks, each point or at least a collection of points has to be classified. In recent past, several overviews and reviews were given on PC segmentation for example by Nyguyen and Le,[12] Guo et. al,[7] Grilli et al.,[13] and Xie et al.[14] Fan et al. propose the SCF-Net which uses spatial learning features.[15] Gong et al.[16] use the neighborhood co-occurrence matrix in combination with the Kullback–Leibler divergence. Lei et al.[17] use a fuzzy spherical kernel and propose the SegGCN net for efficient real-world data segmentation. Another possibility is segmenting PC data in 2D after projection. One approach is given by Lyu et al.[18] Furthermore, they give a detailed overview of PC segmentation methods until 2020. Kellner et al. analyze and compare multiple projection views and test it on SemanticKITTI. To overcome the problem of costly annotating 3D data, Mingmei et al.[19] propose a semi-supervised network called SSPC-Net. 3D data suffers from boundary problems. Tang et al.[20] face it by using contrastive boundary learning (CBL). Wang et al.[21] use a semantic affine transformation (SemAffiNet) to use global structural knowledge in local parts and Thomas et al.[22] present Kernel Point Convolution (KPConv) that can be used for PC segmentation and classification.

3.2 Strategy for I4C

As part of I4C, it is planned to use the KPConv[22] net for the first segmentation tests. The airborne-based data has around 100 million points per 500 $m \times 500$ m. For annotation, crop size of 100 $m \times 100$ m is used. For segmentation the first tests will be carried out with 50 $m \times 50$ m crop size. The point density is planned to be adjusted depending on available hardware resources. Moreover, it is planned to analyse the annotation accuracy to better understand the accuracy of the predictions. First quality assurance tests showed that two people cropped buildings with different accuracies and adjoining structures were assigned Unknown or Building. Another artefact is that there are points within buildings due to LiDAR penetrations through windows. One person annotated it as Unknown, the other as Building. Those annotation errors must be reduced or at least quantified to evaluate the prediction quality. Apart from the constant cop size approach, it is planned to vary the crop size in combination with downsampling to keep the number of points roughly equal. For big classes like Building, a lower density is required than for small classes like cars. Apart from the overall dimension of the class, also the dimension of the minimum distinctive feature is important. An example is a distinction between doors and walls. The distinctive feature is the door handle which is roughly 20 cm in dimension. However, a minimum number of points per door handle is required. Therefore, the minimum point distance should be for example at least 5 cm. In case of additional RGB information, the point density could be lower. In some cases like a car, the position relative to the street can be decisive for prediction. If transferring these examples to the classes of I4C, the following approximated values result which are listed in Table 1. The implementation and testing of the above concepts are part of the ongoing work in the I4C project. The performance of the segmentation, using the approximated cropping sizes and point density will be investigated. In the following, further approaches to enhance the segmentation performance with limited real-world data is given.

3.3 Data augmentation

To overcome the problem of small and unbalanced data sets, augmentation methods can be used as used for 2D images.[23] Similar to image augmentation, color and geometry augmentation is possible. Different from multiple images, the overall 3D data is in the same coordinate system and is, therefore, linked in space. This means that by using different cropping methods, including rotation and translation, a lot of scenes can be created. If the RGB information is used in the segmentation process, color augmentation can stabilize the network and improve the performance on structures similar in shape but different in color. Besides, the lighting conditions may vary between the two seasons but also within one drive or flight. Once the data is annotated, shifting 3D structures and placing them in a different location, with a different orientation, and a different scale is possible. Since the surfaces are known, the objects can be placed on realistic surface types. This can help to increase the number of rare classes within the data set and make the network robust to correctly detect structures in different scenes. Lastly, different instances can be merged to create a new instance. For example, parts of two buildings can

Table 1: Approximated minimum crop size for enough context per class, required point density based on feature size, and estimate if color information and 2D images are required for correct prediction.

Class	Min. Crop size	Distinctive feature	Min. density	RGB	2D prediction
Unknown	-	-	-	-	-
Building	20 m	Openings, roof	10 cm	-	-
Wall	10 m	Joints	1 cm	x	
Conifer	10 m	Color, branches, leaves	2.5 cm	x	-
Deciduous tree	10 m	Color, branches, leaves	1 cm	x	-
Shrub	5 m	Color, roughness	1 cm	x	-
Hedge	10 m	Color, roughness	1 cm	x	-
Grass	10 m	Color	-	x	x
Other vegetation	10 m	Exclusion process	-	x	-
Sealed surface	10 m	Color (asphalt, concrete)	-	x	x
Cobbles	10 m	Color (stone, joint)	-	x	x
Open Surface	10 m	Color (soil, sand)	-	x	x

create a new building. This requires an algorithm that ensures realistic results. Apart from adding structures, removing instances or parts of structures can help. Removing the side of a structure facing away from the sensor is a realistic scenario, that the network must be capable of dealing with.

3.4 Synthetic data

Apart from data augmentation, a network can be trained without having real PC data available. The solution is the use of synthetic data which is increasingly applied in the recent research. Due to the field of gaming and animation a big number of synthetic data and approaches to create realistic data is available and offers great potential to be used to solve real-world problems. Xiao et al. use the knowledge transfer from synthetic data. For this purpose they create a PC translator (PCT) to "mitigate the gap between synthetic and real PCs".[24] Chen et al.[25] present a pipeline to generate synthetic aerial photogrammetry PCs which can be used to overcome imbalanced real data sets. The synthetic data set called STPLS3D covers more than 16 km^2 and 18 classes. Griffiths et al.[26] present an open data set called SynthCity which is a synthetic PC with color and 9 different labels. Wu et al.[27] propose a domain-adaption training pipeline to overcome the gap from synthetic to real-world data. Xu et al.[28] show the benefit of using synthetic data for instance segmentation by replacing the original feature extractor of the SGPN framework. Thereby, the hard examples are more focused by the network. Apart from invented scenes, some cities already created a 3D model of their cities completely or partially.

4. CONCLUSION

As part of this paper, an overview of current research on annotation, augmentation, synthetic data, and segmentation of PC data was given. The goals and background of the project I4C were presented. The specifications of the test areas and the PC data was specified and problems such as artefacts were identified. Merging the data into one PC due to the different perspectives was supposed. The advantage of using data from different seasons such as the distinction of Conifer and Deciduous tree was listed. For annotation, possible tools and a strategy for an efficient sequence were given. Different cropping options were discussed. Here, it was shown that using an equidistant grid is the easiest approach. Lastly, it was decided to use KPKonv for the upcoming segmentation tests. As preparation, for each class, the sufficient cropping size, the distinctive features, the minimum required PC density, and the need of RGB information and prediction in 2D images was approximated. The future work will analyze the made concepts and considerations.

5. FUTURE WORK

The future work will include adapting the KPConv network to the proposed data set and classes. The data set will be annotated based on the developed strategy within this paper. Testing further networks or developing a new network architecture is also possible. A further goal is the automatic derivation of leaf density and the area ratio of sealed surface and joints e.g. for Cobbles. The leaf density is important for water storage and wind interaction. The joints in cobbles determine how much water can seep into soil in the case of a heavy rain event. Apart from our work, more research on efficient annotation and the development of improved visualisation and annotation tools are necessary. Using synthetic data to enrich small data sets will also be important in the future to enable rapid adaptation to the challenges in the future.

ACKNOWLEDGMENTS

This work is part of the project "Intelligence for Cities" (I4C) which is funded by the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV).

REFERENCES

[1] Williams, K., Olsen, M., Roe, G., and Glennie, C., "Synthesis of transportation applications of mobile lidar," *Remote Sensing* **5**(9), 4652–4692 (2013).

[2] "Google earth version 9.168.0.0 - webassembly. herdern freiburg, germany: Google geobasis-de/bkg(©2009)," (2022).

[3] "Google earth version 9.168.0.0 - webassembly. freiburg, germany: Google geobasis-de/bkg(©2009)," (2022).

[4] Wang, B., Wu, V., Wu, B., and Keutzer, K., "Latte: Accelerating lidar point cloud annotation via sensor fusion, one-click annotation, and tracking."

[5] Franzluebbers, A., Li, C., Paterson, A., and Johnsen, K., "Virtual reality point cloud annotation," in [*2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*], 886–887, IEEE, [S.l.] (2022).

[6] Bacim, F., Nabiyouni, M., and Bowman, D. A., "Slice-n-swipe: A free-hand gesture user interface for 3d point cloud annotation," in [*IEEE Symposium on 3D User Interface 2014*], Lécuyer, A., Lindeman, R. W., and Steinicke, F., eds., 185–186, IEEE, Piscataway, NJ (2014).

[7] Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., and Bennamoun, M., "Deep learning for 3d point clouds: A survey," *IEEE transactions on pattern analysis and machine intelligence* **43**(12), 4338–4364 (2021).

[8] Yan, Z., Duckett, T., and Bellotto, N., "Online learning for 3d lidar-based human detection: Experimental analysis of point cloud clustering and classification methods," *Autonomous Robots* (2019).

[9] Ibrahim, M., Akhtar, N., Wise, M., and Mian, A., "Annotation tool and urban dataset for 3d point cloud semantic segmentation," *IEEE Access* **9**, 35984–35996 (2021).

[10] Amazon Web Services, "New – label 3d point clouds with amazon sagemaker ground truth — amazon web services," (2020).

[11] Daniel Girardeau-Montaut, "Cloudcompare: 3d point cloud and mesh processing software: Open source project," (2022).

[12] Nguyen, A. and Le, B., "3d point cloud segmentation: A survey," in [*RAM 2013*], 225–230, IEEE, [Piscataway, N.J.] (2013).

[13] E. Grilli, F. Menna, and Fabio Remondino, "A review ofpoint clouds segmentation and classification algorithms," *undefined* (2017).

[14] Xie, Y., Tian, J., and Zhu, X. X., "Linking points with labels in 3d: A review of point cloud semantic segmentation," *IEEE Geoscience and Remote Sensing Magazine* **8**(4), 38–59 (2020).

[15] Fan, S., Dong, Q., Zhu, F., Lv, Y., Ye, P., and Wang, F.-Y., "Scf-net: Learning spatial contextual features for large-scale point cloud segmentation," 14504–14513 (2021).

[16] Gong, J., Ye, Z., and Ma, L., "Neighborhood co-occurrence modeling in 3d point cloud segmentation," *Computational Visual Media* **8**(2), 303–315 (2022).

[17] Lei, H., Akhtar, N., and Mian, A., "Seggcn: Efficient 3d point cloud segmentation with fuzzy spherical kernel," 11611–11620 (2020).

[18] Lyu, Y., Huang, X., and Zhang, Z., "Learning to segment 3d point clouds in 2d image space," 12255–12264 (2020).

[19] Mingmei Cheng, Le Hui, Jin Xie, and Jian Yang, "Sspc-net: Semi-supervised semantic 3d point cloud segmentation network," *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(2), 1140–1147 (2021).

[20] Tang, L., Zhan, Y., Chen, Z., Yu, B., and Tao, D., "Contrastive boundary learning for point cloud segmentation."

[21] Wang, Z., Rao, Y., Yu, X., Zhou, J., and Lu, J., "Semaffinet: Semantic-affine transformation for point cloud segmentation."

[22] Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., and Guibas, L. J., "Kpconv: Flexible and deformable convolution for point clouds," 6411–6420 (2019).

[23] Negassi, M., Wagner, D., and Reiterer, A., "Smart(sampling)augment: Optimal and efficient data augmentation for semantic segmentation," *Algorithms* **15**(5), 165 (2022).

[24] Xiao, A., Huang, J., Guan, D., Zhan, F., and Lu, S., "Transfer learning from synthetic to real lidar point cloud for semantic segmentation," *Proceedings of the AAAI Conference on Artificial Intelligence* **36**(3), 2795–2803 (2022).

[25] Chen, M., Hu, Q., Hugues, T., Feng, A., Hou, Y., McCullough, K., and Soibelman, L., "Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset."

[26] Griffiths, D. and Boehm, J., "Synthcity: A large scale synthetic point cloud."

[27] Wu, B., Zhou, X., Zhao, S., Yue, X., and Keutzer, K., "Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud," in [*2019 International Conference on Robotics and Automation (ICRA)*], 4376–4382, IEEE, [Piscataway, NJ] (2019).

[28] Xu, Y., Arai, S., Tokuda, F., and Kosuge, K., "A convolutional neural network for point cloud instance segmentation in cluttered scene trained by synthetic data without color," *IEEE Access* **8**, 70262–70269 (2020).