ROOFN3D: DEEP LEARNING TRAINING DATA FOR 3D BUILDING RECONSTRUCTION

Andreas Wichmann, Amgad Agoub, Martin Kada

Institute of Geodesy and Geoinformation Science (IGG), Technische Universität Berlin,
Straße des 17. Juni 135, 10623 Berlin, Germany
firstname.lastname@tu-berlin.de

Commission II, WG II/6

KEY WORDS: 3D Database, Machine Learning, 3D Building Models, Classification, Segmentation, Reconstruction

ABSTRACT:

Machine learning methods have gained in importance through the latest development of artificial intelligence and computer hardware. Particularly approaches based on deep learning have shown that they are able to provide state-of-the-art results for various tasks. However, the direct application of deep learning methods to improve the results of 3D building reconstruction is often not possible due, for example, to the lack of suitable training data. To address this issue, we present RoofN3D which provides a new 3D point cloud training dataset that can be used to train machine learning models for different tasks in the context of 3D building reconstruction. It can be used, among others, to train semantic segmentation networks or to learn the structure of buildings and the geometric model construction. Further details about RoofN3D and the developed data preparation framework, which enables the automatic derivation of training data, are described in this paper. Furthermore, we provide an overview of other available 3D point cloud training data and approaches from current literature in which solutions for the application of deep learning to unstructured and not gridded 3D point cloud data are presented.

1. INTRODUCTION

In recent years, machine learning has been extensively studied with the aim of automatically generating models from data. For this purpose, several machine learning methods have been developed that do not simply memorize examples but are able to automatically recognize patterns and rules in the training data. Particularly approaches based on deep learning have achieved excellent results for different applications. For classification tasks of images, for example, deep learning methods using convolutional neural network (CNN) architectures have become a standard framework in the last few years, as their results are already comparable or even better than from human experts (Cireșan et al., 2012; Krizhevsky et al., 2012).

While CNNs have been a great success for images, they have been, however, less successful for 3D point clouds. The reasons for this are manifold, but they can be mainly attributed either to the lack of publicly available training data or to the specific properties of point clouds: (i) Point cloud data is unstructured and not gridded with varying point density; (ii) The volume of three-dimensional point cloud data is often significantly higher than that of two-dimensional images; (iii) Besides intensity, there is often no radiometric information (e.g., color) available.

Although many different solutions have been already proposed in recent years to apply deep learning to 3D point clouds, a major drawback that still remains is the lack of publicly available training data that can be used to train neural networks. Since machine learning techniques usually require a large amount of training data, this issue is very crucial for carrying out research in this area. While there is a large amount of training data for images available (e.g., ImageNet (Deng et al., 2009), MNIST (Deng, 2012), CIFAR10/CIFAR100 (Krizhevsky, 2009)), the amount of training data for 3D point clouds is comparatively small.

By evaluating a large number of publicly available 3D point cloud training datasets for machine learning, it became clear that a good basis for traditional classification tasks is already available but that the number of classes is generally still quite limited. Particularly for buildings, which play for most applications in urban areas an essential role, we discovered a shortage in the set of available datasets. According to our knowledge, there is currently no 3D point cloud training dataset publicly available that provides distinct classes for buildings. However, many applications require a fine subdivision of the building class, for example, to distinguish between different roof types or to recognize certain roof structures. In order to close this crucial gap, we have developed an automatic workflow in which building points of an aerial LiDAR dataset are processed so that they can be used to train deep neural networks in the context of 3D building reconstruction.

The proposed workflow has been applied to the publicly available New York City (NYC) dataset which consists of over one million buildings. The resulting training dataset is made available through RoofN3D and provides not only geometric information but semantic information as well. Note, RoofN3D provides only training data and is not a benchmark dataset at the moment. The training data is publicly available at https://roofn3d.gis.tu-berlin.de.

The remainder of the paper is organized as follows: First, recent deep learning methods from literature are summarized in which solutions are proposed to overcome the aforementioned issues related to the specific properties of point clouds. In this context, general trends for current and future research directions are pointed out. Afterward, an overview of publicly available 3D point cloud training data for machine learning is presented. Since machine learning methods usually require a large amount of training data, their availability is of great importance. Subsequently, details of the developed workflow and the training data provided through RoofN3D are described. Finally, a conclusion and potential future enhancements concerning RoofN3D are presented.

2. DEEP LEARNING FOR 3D POINT CLOUDS

The automatic recognition of objects is a fundamental task in computer vision. It has recently attracted again the interest of many researchers due to the advancements in artificial intelligence and the development of hardware which enables the implementation and the application of deep neural networks. The results that can be achieved with deep neural networks have already reached a new level for 2D images. Particularly CNNs have proven to be capable of providing state-of-the-art results; see, for example, (Krizhevsky et al., 2012), (Simonyan and Zisserman, 2015), (Szegedy et al., 2015), and (He et al., 2016). Therefore, current research is carried out to apply CNNs not only on 2D data but also on 3D data such as point clouds. The essence of a CNN is, however, the convolutional layer whose parameters consist of a set of kernels (also called filters). These convolutional kernels enable, on the one hand, to share weights in convolutional layers and thus significantly reduce the total number of parameters in a CNN. On the other hand, convolutional kernels always require data in a regular structure as their layer input. Therefore, CNNs cannot be directly applied to 3D point cloud data which consist of a set of unordered points. For further information about CNNs, see, for example, (Goodfellow et al., 2016).

To overcome this issue and to adequately deal with the high volume of three-dimensional point cloud data, solutions have been developed that are based on the conversion of the irregular point cloud data structure to a regular data structure. Some proposed regular representations in the context of CNNs with respect to 3D point cloud data are described in section 2.1. In addition to approaches based on regular representations, further approaches for neural networks have been developed that are able to directly process data represented in an irregular data structure. Thereby, issues that accompany regular data structures are automatically avoided. Some details of these irregular representations are summarized in section 2.2.

2.1 Regular Representations

A solution that has been proposed to represent 3D point cloud data in a regular structure, so that they are suitable for CNNs, is the multi-view representation. In this approach, multiple 2D views (i.e., images) of a 3D object are generated from different viewpoints. Thereby, the dimension of the 3D input data is reduced for each viewpoint to 2D. An important aspect in this context is how the viewpoints can be determined. Different approaches have been developed for multi-view CNNs in which they are either set empirically (Su et al., 2015) or automatically (Huang et al., 2018). Once a set of 2D images has been captured from different viewpoints, each image is passed through a convolutional network and the results of these networks are subsequently aggregated using a view-pooling layer. Further examples of different multi-view CNNs are presented in (Qi et al., 2016) and (Yi et al., 2017).

Multi-view representations provide the advantage that traditional neural networks for images can be applied with only minor adaptions. However, it needs to be considered that the conversion is usually accompanied by loss of information due to the limited number of viewpoints and the fact that each viewpoint can only partially represent a 3D object. Particularly occlusions can exacerbate this problem. All this might result in inconsistencies during the reconstruction of surfaces in 3D space.

Another group of regular representations, which keep the dimensions of the input data, is the volumetric representation. In the last years, different variants of volumetric representations have been proposed for CNNs. A well-known representative is, for example, the voxel grid. In a voxel grid, the three-dimensional space is discretized into a regular grid and each resulting voxel is assigned a value based on the points within the voxel. Depending on the object to be detected, the resolution of the voxel grid needs to be adjusted. A special type of a voxel grid, which is commonly in use, is the occupancy grid (Thrun, 2001). The special characteristic of an occupancy grid is that each voxel is assigned only the value occupied or unoccupied, depending on the presence or absence of data. Some examples in which different variants of voxel grids have been used as input for CNNs are VoxNet (Maturana and Scherer, 2015), 3D ShapeNets (Wu et al., 2015), volumetric CNNs (Qi et al., 2016), and SEGCloud (Tchapmi et al., 2017).

A major advantage of representing 3D point clouds in a voxel grid is that already existing CNN architectures for images can be generally applied with only few adaptions. However, since objects are represented in 3D point cloud data only on their surface, the input data easily become unnecessarily voluminous due to the large number of unoccupied voxels. An illustration of this so-called sparsity problem of 3D data in occupancy grids is shown in Figure 1 for different resolutions. In order to overcome this problem, while keeping the spatial information about the 3D shape, other regular data structures have been developed such as deep data structures and convolutional filters that can work on them.

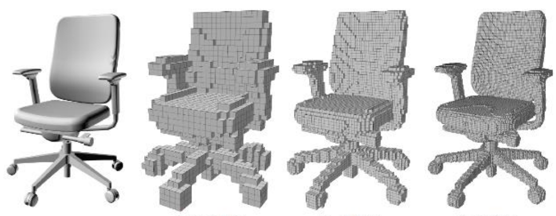

Figure 1. The percentage of occupied grids in different grid resolutions. The percentage at resolution 30^3 is 10.41% and reduced to 5.09% and 2.41% at resolution 64^3 and 128^3, respectively. (Li et al., 2016)

In deep data structures, special focus is placed on the representation of unoccupied voxels. For this, only occupied voxels are recursively divided up to a certain limit. Thereby, the amount of data to represent unoccupied voxels is significantly reduced. To illustrate the impact, a comparison between a voxel grid and a deep data structure is shown in Figure 2. Thus, deep data structures enable in practice the use of a higher grid resolution compared to occupancy grids. Some examples in which deep data structures have been used are given in (Riegler et al., 2017), (Klokov and Lempitsky, 2017), and (Wang and Posner, 2015).

A general drawback of regular data structures is that the resolution to represent 3D data is generally limited. Therefore, the conversion of irregular to regular 3D data structures is usually accompanied with loss of information. Furthermore, some information in the input data such as symmetry and roundness cannot simply be captured when regular data structures without very high resolutions are used. Designing machine learning architectures that can directly process irregular 3D data can help to overcome these challenges and also the data sparsity problem.

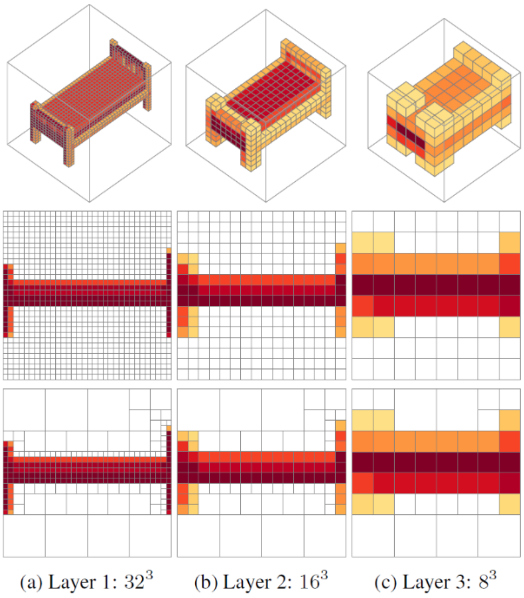

Figure 2. An object displayed in voxels in different resolutions (top row) once represented in a voxel grid (middle row) and once represented in a deep data structure (bottom row). Occupied cells are colored and indicate activation, while unoccupied are colored white. (Riegler et al., 2017)

2.2 Irregular Representations

In addition to regular representations, other solutions have been proposed for deep learning with respect to 3D point clouds based on irregular representations. In (Fang et al., 2015), for example, shape descriptors are used which enable the identification of a 3D object as a member of a category based on a concise and informative representation. Thereby, 3D input meshes are represented in 2D without losing relevant information. Figure 3 shows the proposed workflow for the automatic derivation of shape descriptors using deep learning.

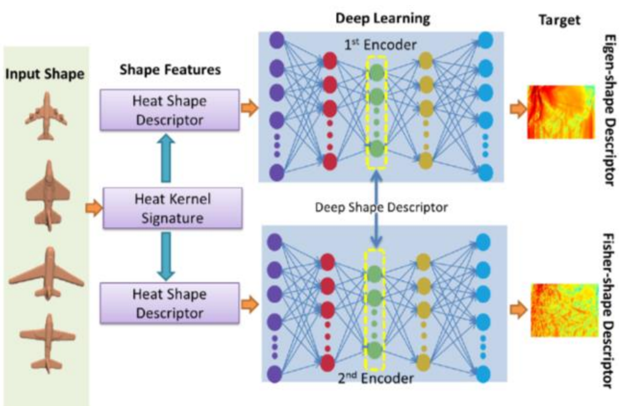

Figure 3. The automatic extraction workflow of shape descriptors presented in (Fang et al., 2015).

In (Guo et al., 2015), a compact mesh representation is learned by extracting multiple geometric low-level features. Based on the extracted features, a 2D tensor is constructed that serves as input for a CNN model. Using a 2D tensor as input for the network instead of combining features by concatenation enables hierarchal aggregation and interaction of the extracted features through the CNN network weights.

Another approach that enables the use of 3D point clouds as input for a neural network has been proposed in PointNet (Qi et al., 2017a). For this, it has the following two special components: (i) a permutation invariant module and (ii) a 3D spatial transformation module. A permutation invariant module can be considered in PointNet as a shared weighted multi-layer perceptron with max pooling aggregation. An advantage of using symmetric functions such as max functions is that networks are able to be invariant to permutations of their input members. A spatial module is a deep learning network in which the input is spatially transformed to a canonical form. The canonical form can be considered in this context as the spatial pose of an object which improves the performance of the deep learning network. The transformation into the canonical form enables the network to be invariant to spatial translations.

Compared to traditional CNNs or 3D CNNs that benefit from hierarchical feature learning, global feature learning embeddings for one point and a global vector for all the points are used in PointNet. Therefore, the effect of a local context is generally weak when applying PointNet. However, the positive effect of leveraging neighborhood relations in a deep learning model have already been shown in some approaches such as (Gressin et al., 2013) and (Weinmann et al., 2015). In order to address this issue, a successor of PointNet called PointNet++ (Qi et al., 2017b) has been developed. In this deep learning architecture, spherical point neighborhoods of large scenes are extracted and the PointNet architecture is applied on each part in a hierarchical way. Thus, PointNet++ can be considered as a hierarchical version of PointNet, which takes neighborhood information in its deep learning architecture into account.

A group of machine learning models that can be used to derive irregular 3D data from different 2D and 3D data is the generative adversarial network (GAN) (Goodfellow, 2016). In a GAN architecture, point sets are encoded and decoded using machine learning techniques. For this, some recent approaches follow the concepts of PointNet by designing an autoencoder that is invariant to permutations and spatial transformations. In (Achlioptas et al., 2018), for example, such an autoencoder is used in a GAN to derive 3D point sets according to thematic classes based on a 3D point set.

3. AVAILABLE 3D POINT CLOUD TRAINING DATA

In machine learning, massive training data is usually needed for learning a model such as a neural network. Thus, their availability is nowadays of high importance. A challenge for the provision of such training data is that a sufficient amount of training data needs to be provided. Otherwise, the trained model would tend to overfit the data in the sense that specific relationships in the training data are identified that do not hold in general. Another important aspect of the availability of training data is that it can be also used to evaluate machine learning models. Many machine learning models are constructed as very complex mathematical models, which can make a theoretical evaluation of their performance very challenging and controversial. However, a practical approach to address this challenge is to empirically evaluate machine learning models within a common framework based on publicly available data. Empirical evaluations provide researchers the ability to compare and evaluate the performance of their models according to set standards. Early publicly available datasets have been proven to be of great value to research. In (Sabour et al., 2017), for example, the MNIST (Deng, 2012) dataset was used to prove the effectivity of dynamic routing between capsules. Generally, it is noticeable that many research communities are nowadays interested in acquiring and producing such data in 3D. With ShapeNet (Chang et al., 2015) and ModelNet (Wu et al., 2015), for example, datasets of 3D models from common 3D objects are publicly available. Some further publicly available 3D datasets are listed in the following paragraph.

There are, nowadays, many 3D data available that were captured in urban areas with the help of mobile laser scanning. This includes, for example, data from street sections such as the Paris-rue-Madame dataset (Serna et al., 2014), which consists of about 10 million points and 27 classes, or the Paris-rue-Soufflot dataset (Hernándes and Marcotegui, 2009), which consists of about 20 million points and 6 classes. But also data of larger urban areas are available such as the IQumulus & TerraMobilita Contest dataset (Vallet et al., 2015), which consists of about 300 million points and about 80 classes, and the Semantic3D.Net dataset (Hackel et al., 2017), which consists of over four billion manually labelled points and 8 classes.

The provided datasets listed in the previous paragraph are generally suitable for traditional tasks such as recognition and classification in the urban context. However, for the purpose of 3D building type classification and reconstruction, these datasets would be not sufficient because they do not provide distinct subclasses within the building class which indicate the building parts. The Oakland dataset (Munoz et al., 2009), which consists of about 1.6 million points and 44 classes, can also only partially overcome this issue. It contains classes for building parts and a specific label for roofs but it does not provide an explicit class distinction between different roof types.

To the best of our knowledge, there is currently no publicly available dataset with semantic roof types for the purpose of learning different roof types and 3D building reconstruction based on machine learning models. By presenting an automatic workflow that can provide such training datasets and by providing the results of already processed data on RoofN3D, we aim to close this crucial gap and we would be pleased about the use of this data in other research work.

4. TRAINING DATA PREPARATION FRAMEWORK

For the provision of massive 3D data and further information that are needed to train deep neural networks in the context of 3D building reconstruction, an automatic framework has been developed. The automatic framework has been designed to be capable of efficiently processing very large point clouds with billions of points. Thereby, huge training datasets consisting of different building classes and a large number of instances for each class can be generated in fairly short time. An overview of the whole framework is illustrated in Figure 4. It consists of the following two major steps: (i) extraction of building points and (ii) derivation of building information.

In the building point extraction step, aerial point clouds and building footprints are used as input data for the developed framework. Both types of data are nowadays already publicly available for several cities, states, and countries around the world (e.g., New York City (USA), Philadelphia (USA), Toronto (Canada), Vancouver (Canada), Cambridge (UK), North Rhine-Westphalia (Germany), Thuringia (Germany), The Netherlands, etc.). Due to the large number of points in the provided point clouds, a direct extraction of building points is usually not feasible in a reasonable time if, for example, only building footprints in combination with a ray casting algorithm are used to solve the point-in-polygon problem. Therefore, point clouds and building footprints are not directly intersected with each other to determine the building points but multiple patches are first generated for each point cloud. The resulting patches are then intersected with those building footprints that are located in the area of the point cloud. In this way, all patches are identified that feature an overlap with a building. Afterward, the relevant patches are exploded and all those points are classified as building points that are located within a building footprint of the processed area. The classified points and the building footprints are finally stored in the RoofN3D database.

The derivation of building information based on building points follows the principles of common data-driven building reconstruction approaches and consists of the following three

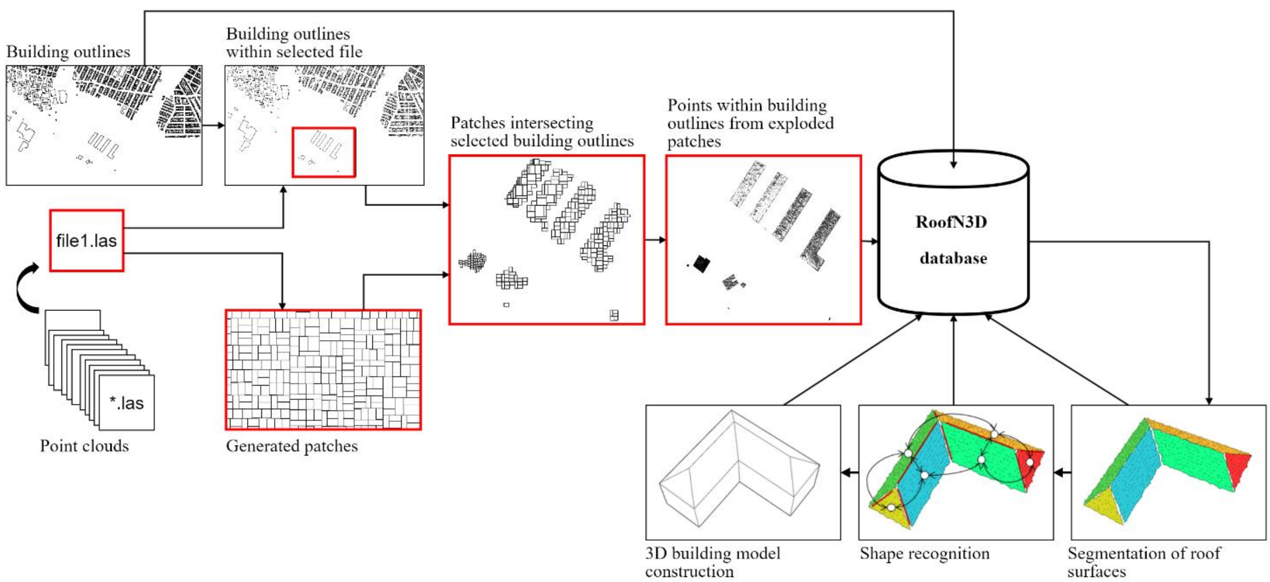

Figure 4. Framework for the automatic preparation of data that is needed to train neural networks in deep learning methods.

sub-steps: (i) segmentation of roof surfaces, (ii) shape recognition, and (iii) 3D building model construction. Since the results of each sub-step can be used to train neural networks for different tasks, they are stored in the RoofN3D database.

For the segmentation of roof surfaces on the basis of building points, sub-surface growing as introduced in (Kada and Wichmann, 2012) is applied with some extensions. In contrast to the well-known surface growing method, segments are enabled during the sub-surface growing procedure to grow below the surface. As a result, segment patches that belong to the same roof surface, but are disconnected by roof superstructures, are merged together. Consequently, symmetries are implicitly modeled and the number of primitives is reduced of which a building with complex roof structure is comprised of. Furthermore, segments of connected building components (e.g., dormer and base roof) feature actual intersections so that gaps between them are automatically closed. The advantages of sub-surface growing support the identification of roof structures in the subsequent shape recognition step and make their detection more robust than with conventional surface growing.

In order to further improve the segmentation results, sub-surface growing has been extended by a reassignment method. In this method, all points assigned to a segment are reassigned to a neighboring segment in cases where they would better fit the neighboring segment in terms of distance and normal vector direction. The reassignment process is carried out once after surface growing and once after sub-surface growing. In addition to the improved assignment of points that already belong to a segment, sub-surface growing has also been extended in such a way that already segmented roof surfaces are further enriched with points not previously assigned to any segment. For this, a point in the set of unassigned points is assigned to its closest segment after the reassignment process has been completed if its distance to the segment is within a certain tolerance. Both described extensions are suitable to improve the segmentation result of sub-surface growing.

Once planar roof surfaces have been determined, all segments are represented as nodes and their adjacency relationships as connecting edges in a so-called roof topology graph (RTG). For the recognition of certain roof shapes in the RTG, a graph grammar has been developed in which production rules are defined that represent possible graph transformations. Thereby, the search for predefined roof shape models does not need to be performed directly on the input data but it can be carried out on higher-level information in the so-called topology space. Thus, the robustness of traditional model-driven recognition approaches is maintained while reducing the search effort and the computational time.

Each production rule of the developed grammar consists of two graphs representing its left-hand side (LHS) and its right-hand side (RHS). If a production rule is applied to the RTG, all occurrences of the LHS in the RTG are first identified by a labeled graph matching algorithm and then replaced by its RHS. The production rules have been essentially designed in such a way that adjacent nodes and their connecting edges, which represent lower-level information, are aggregated to a single node which represents higher-level information about the building. For example, two connected nodes that both represent sloped segments, whose segment normals point in the horizontal plane in the opposite direction, and which have an intersection line of a certain minimum length are aggregated to a single node that represents the semantic information of a saddleback roof. With each aggregation, semantic information is added to the RTG. By applying several production rules of the graph grammar to the RTG, higher-level information about the building to be reconstructed is derived. Thereby, the knowledge of the building structure including the building parts becomes available. To ensure that unnatural shapes are automatically discarded, already derived building knowledge is incorporated during the application of production rules. Due to the expressive power of the applied grammar, not only geometric information but also extensive semantic information can be provided.

Finally, 3D building models are constructed based on half-space modeling as introduced in (Kada and Wichmann, 2013) and adjusted based on the divisive clustering techniques introduced in (Wichmann and Kada, 2014) to support natural building characteristics (e.g., symmetry, co-planarity, orthogonality). By utilizing half-space modeling, buildings are represented within the proposed framework as closed solids to guarantee that all building models are topologically correct and that they do not feature any unwanted gaps or intersections.

5. ROOFN3D

In order to close the lack of publicly available 3D point cloud data that are suitable to train neural networks for different tasks in the context of buildings, we present RoofN3D. It provides a platform for the distribution of 3D training data that result from the application of the presented training data preparation framework to various data. As a first step, the publicly available New York City dataset of the NYC Open Data Portal (https://opendata.cityofnewyork.us) has been processed and suitable parts of the resulting training data are made available via RoofN3D. Some information about the New York dataset are summarized in section 5.1 and further details about the provided data on RoofN3D are given in section 5.2. Note, further datasets will be processed in the future and their results will be added to RoofN3D. The training data is available at https://roofn3d.gis.tu-berlin.de.

5.1 New York Dataset

The building footprint dataset of New York is part of the planimetrics geodatabase and used by the NYC Department of Information Technology and Telecommunications (DoITT) GIS group to maintain and distribute an accurate base map for NYC. They are derived from images of the 2014 New York Statewide Flyover, which includes raw imagery collected to support the generation of 0.5 feet ground sampling distance natural color imagery. The provided building footprints represent the perimeter outline of each building. Divisions between adjoining buildings are determined by tax lot divisions. The estimated positional accuracy for 95% of the data is ± 2 feet. The whole dataset consists of more than one million building footprints.

The LiDAR point clouds of New York are provided by the U.S. Geological Survey (USGS). They have been captured from 08/2013 to 04/2014 and cover an area of 1,009.66 km². The average density of the point clouds is about 4.72 points/m².

5.2 RoofN3D Data

The available data from RoofN3D currently consist of the results of the presented training data preparation framework that has been applied to the New York dataset. The New York dataset has been selected because it covers a large area, which can help to avoid an overfitting of classifiers. An overview of the underlying architecture of RoofN3D and the available information about the buildings are shown in Figure 5. As illustrated, for each building in RoofN3D, a building footprint, all building points therein, segmentation results, semantic information, and geometric information are provided. Please note, the structure of RoofN3D is not fixed and might be adapted according to future needs.

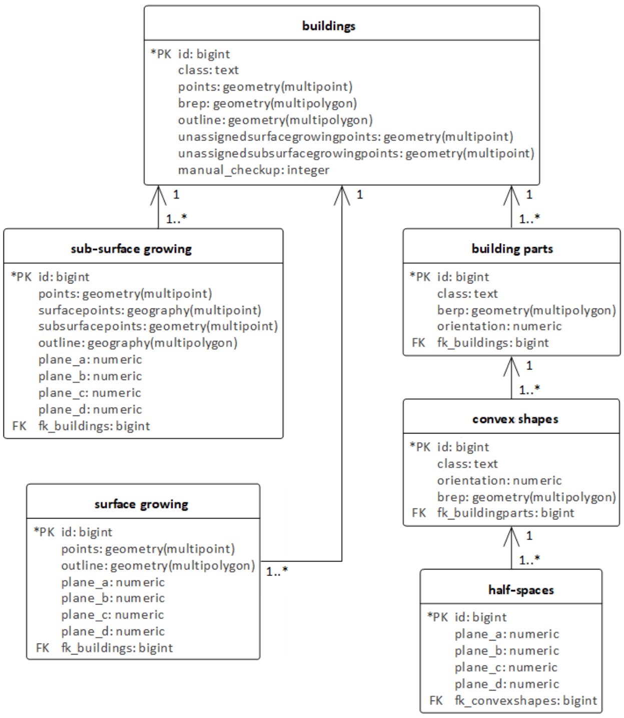

Figure 5. The architecture of RoofN3D and the provided information about a building.

The provided data can be used in various ways. For example, data resulting from the segmentation process can be used to train semantic segmentation networks. For this, each extracted segment provides the information about its assigned points, the outline, and the plane equation which best fits the assigned points. We have limited the segmentation process to planar areas because most roof surfaces can be described with sufficient accuracy by planar faces. This is for most buildings, particularly in residential areas, the case and useful because arbitrary shapes usually require computationally intensive surface fitting procedures (Wang, 2013). Since sub-surface growing has been implemented as an extension of the well-known surface growing method, segments of both surface growing and sub-surface growing are provided. For the latter, a distinction is made between surface points and virtual sub-surface points which have been not initially captured but introduced to close unwanted gaps in a roof surface. In addition to the points that have been assigned to a segment, also all unassigned building points are provided which do not belong to any segment according to the applied surface growing and sub-surface growing method. The outline of a segment has been determined by projecting all segment points onto the plane of the segment and by performing the alpha shape algorithm presented in (Edelsbrunner et al., 1983).

To learn the structure of buildings with deep learning methods, the applied grammar of the shape recognition step has been designed in such a way that structural information about the buildings are derived. This includes the information about the building parts that compose the building. Depending on the complexity of the building, the number and the shape of the building parts can be very diverse. In order to provide further information about those building parts that have a concave shape, information about convex components that compose the concave building part are provided. To reduce the processing effort for training a deep neural network, also information about the orientation of building parts and their convex components are provided.

Another important task that can be approached by means of neural networks and the provided training data is the learning of the geometric construction of 3D building models. For this, RoofN3D provides boundary models not only for the whole building but also for its building parts. These boundary models are derived from the conversion of the closed solid that results from the applied automatic reconstruction method. Thereby, it can be guaranteed that the boundary models are always closed and represented as 2-manifold polyhedrons. Note, due to the lack of ground level height information, all building models have been extruded to the same ground height (i.e., 0 m). In addition to boundary models, information about half-space models are provided via RoofN3D. For this, the plane equations of the hyperplanes that define the half-spaces of the roof of a convex building component are provided. By applying the Boolean intersection operator to these half-spaces, the roof shape of the convex component is defined. Furthermore, the roof shape of a building part can be derived by applying the Boolean union operator to the half-space representations of all convex components that compose the building part. Analogous, the roof shape of the whole building is given by uniting all half-space models of those building parts that compose the building. If the shape of the whole building with extruded facades is needed, the latter half-space model needs to be intersected once with a half-space having a horizontal hyperplane whose normal vector is pointing downwards and once with the provided building footprint formulated as a half-space model.

The aforementioned information are offered via RoofN3D for different types of roofs. First, the number of different roof types is limited and only cover simple shapes such as gable roofs, two-sided hip roofs, pyramid roofs, etc. However, more complex roof types will be added over time.

6. CONCLUSION AND OUTLOOK

The training dataset available on RoofN3D provides aerial LiDAR data and building information that can be used to train deep neural networks for different tasks in the context of 3D building reconstruction. The training dataset has been recently published and will be extended in the future according to the needs of the research community. This includes, for example, the addition of further buildings with the same or other roof shapes. For the latter, buildings with complex roof shapes are of particular interest. Furthermore, it is planned to carry out a quality assessment and to continuously improve the offered data. This is necessary because the training data was generated with an automatic process. We hope that the 3D training data on RoofN3D can be used in the future to further improve automatic 3D building reconstruction approaches and their results with various methods of machine learning.

ACKNOWLEDGEMENT

We are grateful to the city of New York for providing building footprints through the New York City Open Data Portal (https://opendata.cityofnewyork.us) and to the U.S. Geological Survey (USGS) for providing LiDAR point clouds of New York.

REFERENCES

Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L., 2018. Learning Representations and Generative Models for 3D Point Clouds. In: *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. ShapeNet: An Information-Rich 3D Model Repository. In: *arXiv preprint arXiv:1512.03012*.

Cireșan, D., Meier, U., Schmidhuber, J., 2012. Multi-Column Deep Neural Networks for Image Classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3642–3649.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A Large-Scale Hierarchical Image Database. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255.

Deng, L., 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research. In: *IEEE Signal Processing Magazine*, 29(6), pp. 141–142.

Edelsbrunner, H., Kirkpatrick, D., Seidel, R., 1983. On the Shape of a Set of Points in the Plane. In: *IEEE Transactions on Information Theory*, 29(4), pp. 551-559.

Fang, Y., Xie, J., Dai, G., Wang, M., Zhu, F., Xu, T., Wong, E., 2015. 3D Deep Shape Descriptor. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2319–2328.

Goodfellow, I., 2016. NIPS 2016 Tutorial: Generative Adversarial Networks. In: *arXiv preprint arXiv:1701.00160v4.*

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. The MIT Press, Cambridge, USA.

Gressin, A., Mallet, C., Demantké, J., David, N., 2013. Towards 3D LiDAR Point Cloud Registration Improvement Using Optimal Neighborhood Knowledge. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 79, pp. 240–251.

Guo, K., Zou, D., Chen, X., 2015. 3D Mesh Labeling via Deep Convolutional Neural Networks. In: *ACM Transactions on Graphics (TOG)*, 35(1):3.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., Pollefeys, M., 2017. Semantic3D.Net: A New Large-Scale Point Cloud Classification Benchmark. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-1/W1, pp. 91–98.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Hernández, J., Marcotegui, B., 2009. Filtering of Artifacts and Pavement Segmentation from Mobile LiDAR Data. In: *ISPRS Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XXXVIII-3/W8, pp. 329–333.

Huang, H., Kalogerakis, E., Chaudhuri, S., Ceylan, D., Kim, V. G., Yumer, E., 2018. Learning Local Shape Descriptors from Part Correspondences with Multi-View Convolutional Networks. In: *ACM Transactions on Graphics (TOG)*, 37(1):6.

Kada, M., Wichmann, A., 2012. Sub-Surface Growing and Boundary Generalization for 3D Building Reconstruction. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3, pp. 233–238.

Kada, M., Wichmann, A., 2013. Feature-Driven 3D Building Modeling Using Planar Halfspaces. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W3, pp. 37–42.

Klokov, R., Lempitsky, V., 2017. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 863–872.

Krivzhevsky, A., 2009. Learning Multiple Layers of Features from Tiny Images. In: *Technical Report, Department of Computer Science, University of Toronto*.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*.

Li, Y., Pirk, S., Su, H., Qi, C. R., Guibas, L. J., 2016. FPNN: Field Probing Neural Networks for 3D Data. In: *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS)*.

Maturana, D., Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 922–928.

Munoz, D., Bagnell, J. A., Vandapel, N., Hebert, M., 2009. Contextual Classification with Functional Max-Margin Markov Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 975–982.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85.

Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L. J., 2016. Volumetric and Multi-View CNNs for Object Classification on 3D Data. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5648–5656.

Qi, C. R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*.

Riegler, G., Ulusoy, A. O., Geiger, A., 2017. OctNet: Learning Deep 3D Representations at High Resolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6620–6629.

Sabour, S., Frosst, N., Hinton, G. E., 2017. Dynamic Routing between Capsules. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*.

Serna, A., Marcotegui, B., Goulette F., Deschaud, J.-E., 2014. Paris-Rue-Madame Database: A 3D Mobile Laser Scanner Dataset for Benchmarking Urban Detection, Segmentation and Classification Methods. In: *Proceedings of the 3rd International Conference on Pattern Recognition, Applications and Methods (ICPRAM)*, pp. 819–824.

Simonyan, K., Zisserman, A., 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 945–953.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going Deeper with Convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.

Tchapmi, L. P., Choy, C. B., Armeni, I., Gwak, J., Savarese, S., 2017. SEGCloud: Semantic Segmentation of 3D Point Clouds. In: *Proceedings of the 5th International Conference on 3D Vision (3DV)*.

Thrun, S., 2001. Learning Occupancy Grids with Forward Models. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No.01CH37180)*, 3, pp. 1676–1681.

Vallet, B., Brédif, M., Serna, A., Marcotegui, B., Paparoditis, N., 2015. TerraMobilita/iQmulus Urban Point Cloud Analysis Benchmark. In: *Computers & Graphics*, 49, pp. 126–133.

Wang, R., 2013. 3D Building Modeling Using Images and LiDAR: A Review. In: *International Journal of Image and Data Fusion*, 4(4), pp. 273–292.

Wang, D. Z., Posner, I., 2015. Voting for Voting in Online Point Cloud Object Detection. In: *Proceeding of the 11th International Conference on Robotics: Science and Systems (RSS)*.

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 105, pp. 286–304.

Wichmann, A., Kada, M., 2014. 3D Building Adjustment Using Planar Half-Space Regularities. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3, pp. 189–196.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920.

Yi, L., Shao, L., Savva, M., Huang, H., Zhou, Y., Wang, Q., Graham, B., Engelcke, M., Klokov, R., Lempitsky, V., Gan, Y., Wang, P., Liu, K., Yu, F., Shui, P., Hu, B., Zhang, Y., Li, Y., Bu, R., Sun, M., Wu, W., Jeong, M., Choi, J., Kim, C., Geetchandra, A., Murthy, N., Ramu, B., Manda, B., Ramanathan, M., Kumar, G., P., P., Srivastava, S., Bhugra S., Lall, B., Häne, C., Tulsiani, S., Malik, J., Lafer, J., Jones, R., Li, S., Lu, J., Jin, S., Yu, J., Huang, Q., Kalogerakis, E., Savarese, S., Hanrahan, P., Funkhouser, T., Su, H., Guibas, L., 2017. Large-Scale 3D Shape Reconstruction and Segmentation from ShapeNet Core55. In: *arXiv preprint arXiv:1710.06104v2*.