Generating Watertight 3D Building Models from Airborne LiDAR Point Clouds using Detection Transformer (DETR)

Lilli Kaufhold[1], Martin Kada[1]

[1] Institute of Geodesy and Geoinformation Science, Technische Universität Berlin, Germany
(lilli.kaufhold, martin.kada)@tu-berlin.de

Keywords: Buildings, 3D, Airborne Laser Scanning (ALS), Deep Learning, Geometry, Reconstruction.

Abstract

This work proposes a method for creating accurate, watertight 3D building models from airborne laser scanning (ALS) point clouds by leveraging a modified Detection Transformer (DETR) architecture. We adapted the DETR architecture to directly predict building planes from point clouds, from which the 3D model can be inferred using Boolean operations of half-spaces. We tested the model on the RoofN3D dataset and achieved a mean angle error of 1.7° for the building planes and a mean point-to-plane distance of 0.16m. Building facets can be detected even in the total absence of representative points, a common challenge in ALS data due to scanning direction and occlusions. By learning higher-level geometric principles, such as favouring 90-degree angles and symmetry, the model is able to adapt to various architectural styles without the need for explicit rules or pre-defined roof archetypes.

1. Introduction

Three-dimensional (3D) city models are increasingly being used in applications such as urban planning, disaster management, cultural heritage conservation, resource optimisation such as the location of solar panels sites and the creation of digital twins for real-time monitoring and simulation (Biljecki et al., 2015, Romero Rodríguez et al., 2017, Gao et al., 2018)). In addition to topographic objects like vegetation, roads, street furniture, and an underlying terrain representation, buildings are often an important part of these models, and the focus here is to capture their geometries. While small areas can be modelled by hand, large-scale models need to be automatically extracted and reconstructed. There exist a number of methods to tackle this problem, as, for instance, discussed in (Haala and Kada, 2010, Buyukdemircioglu et al., 2022).

A common data source for extracting 3D building models is airborne laser scanning (ALS). Although satellite images and TrueOrtho photos provide high-resolution texture, they do not include 3D information. To introduce 3D aspects, additional data sources such as digital surface models (DSMs) would have to be used. Point clouds, generated through photogrammetry or lidar (Light Detection and Ranging), provide 3D data. While in photogrammetric point clouds the buildings are often occluded by trees, ALS has the advantage of being able to penetrate vegetation to map underlying structures. They are also widely available. Depending on whether the images are taken obliquely or from a nadir perspective, vertical surfaces such as building façades may be only partially visible or completely missing, which needs to be taken into account by the reconstruction method.

However, raw point cloud data presents challenges for downstream tasks due to its unstructured nature and complexity. Thus, there is a need to create 3D models that are more manageable and still capture relevant information. Measuring the quality is not straightforward and quality measures should match the intended use of the models (Oude Elberink and Vosselman, 2011). In general, there are several desired properties, including:

- **Accuracy:** The extracted 3D models represent the actual geometries of buildings, ensuring that vertex locations, edges, roof angles, etc. are preserved according to the true physical forms captured by ALS data. Inaccuracies in building dimensions can significantly affect downstream applications, such as urban airflow simulation; for example, (Carpentieri and Robins, 2015) demonstrated that errors in building height can significantly alter simulated wind flow and dispersion.

- **Aesthetics:** Models should maintain the architectural integrity of the original structures. This includes the preservation of design elements such as symmetries, right angles, and other distinctive features. These also depend on the cultural and historical context of the building.

- **Compactness:** Models should be simple, avoiding the use of an excessive number of primitives that make downstream processing tasks computationally expensive. As highlighted in (Yu et al., 2021), overly complex models can hinder real-time rendering in interactive applications and increase computational costs in simulations, which is especially problematic in large-scale urban environments.

- **Geometric Integrity:** Geometric flaws can limit the use in different tools and exchange, and lead to errors in subsequent computations. Thus, models should be watertight, free from self-intersections, and have non-degenerate faces. Furthermore, the surface should form a 2-manifold geometry.

- **Editability:** As some automatically generated models might not meet user requirements, especially in the case of buildings with special or uncommon geometries, it should be easy to adapt and correct the resulting models manually. This means that changes should involve low effort and lead to building models that still fulfil the above-mentioned requirements.

The primary focus of our study is to investigate the feasibility of using the detection transformer (DETR) model to produce

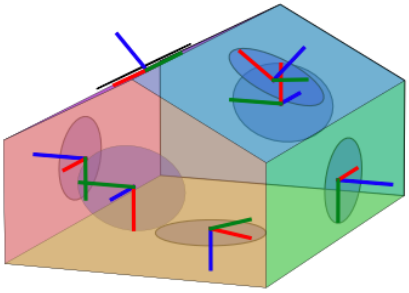

Figure 1. The DETR model predicts a set of planes, which are combined through intersection to generate a 3D building model, following the half-space concept.

accurate plane parameters necessary for building reconstruction. We confirmed this by training the model to generate 3D models for convex buildings that meet the above-mentioned requirements. Specifically, 3D building models are represented in a compact manner using half-spaces (Figure 1). These divide the space into ”inside” and ”outside” and are combined using Boolean operators to create watertight volumes. The approach does not rely on a predefined set of architectural rules but infers them from the training data. This is enabled by the transformer’s attention mechanism, which captures the relationships between all building components that leverage symmetries and orthogonal angles to fill in building facets not represented in the input point cloud.

2. Related Work

Most existing methods try to balance some or all of these requirements. Model-based techniques typically contain explicit restrictions on the allowed shapes derived from architectural structures. This ensures that the resulting 3D models fulfil high-level assumptions, such as matching predefined roof types. In contrast, data-driven or more specifically observation-driven methods focus on aligning the 3D model closely with the input point cloud. This alignment guarantees that the model accurately corresponds to the measurements. In practice, all methods integrate both model-driven and observation-driven elements to create a balanced final model.

2.1 Traditional Methods

An overview of traditional methods can be found in (Haala and Kada, 2010) and (Tomljenovic et al., 2015). These methods often rely on the identification of basic geometric primitives, such as planes. To this end, algorithms such as RANSAC (Schnabel et al., 2007, Li and Shan, 2022, Sun et al., 2024), Hough transform (Tian et al., 2020, Ballard, 1981), and region growing (Vosselman and Dijkman, 2001, Liu et al., 2023) are often used. For a large-scale reconstruction of all buildings in the Netherlands, Peters et al. (Peters et al., 2022) used a decomposition of the building footprints. They then reconstructed the buildings from extracted planes. Planes are merged based on their proximity, with thresholds controlling the merging to scale the model between accuracy and compactness. These approaches do not address symmetry directly; however, since the model relies on the footprint of the building, opposite sides will be as symmetric as the footprint polygon sides.

Instead of using basic primitives, more model-driven methods explicitly model the roof types. In (Li and Shan, 2022), Li and Shan use a nested RANSAC approach which estimates the plane parameters in the inner loop, as well as the full building parameters in the outer loop based on the extracted planes. Here, adherence to predefined models pose hard constraints, so geometric integrity of the resulting models can be guaranteed. In this way complex buildings with missing parts can also be constructed.

Others propose energy-minimisation frameworks that combine data fidelity, smoothness, and complexity penalties (Hu et al., 2021, Li et al., 2023). These approaches typically take a lot of effort to adapt to point clouds acquired in a different way, as the feature extraction has to be tuned manually.

The authors of (Bizjak et al., 2021) detect half-planes that are then intersected and combined using Boolean operations to form a final 3D model, which leads to geometrically valid models, but is not adaptable to user preferences.

2.2 Machine Learning Methods

Supervised machine learning algorithms derive implicit models from labelled training data. An overview of deep learning techniques in the area of building reconstruction can be found in (Buyukdemircioglu et al., 2022). A key challenge is extracting robust features from noisy point clouds. Consequently, several methods replace hand-crafted features with learnt representations (Buyukdemircioglu et al., 2022). For instance, (Soleimani Vostikolaei and Jabari, 2023) classify the types of (single part) building roofs using a conventional neural network into a predefined set of roof types based on optical RGB images and a DSM, and then determine the building parameters, like ridge and eaves height, from a normalized DSM. From the primitive parameters, geometrically valid 3D models can be generated in a rule-based manner in any geometrical representation, but the method is limited to a small set of predefined building types.

Segmentation-based techniques focus on partitioning the input data into meaningful regions that correspond to different building components, such as roof planes. For example, (Kada, 2022) add prediction heads to jointly estimate the plane parameters of each segmented part, thus also replacing the parameter estimation step. Similarly, Li et al. (Li et al., 2024) propose a boundary-aware clustering architecture to segment the point cloud into roof planes, which could then be parametrised to obtain the final 3D model. Others focus on roof edges: (Xu et al., 2024) uses CNNs to extract vectorised roof lines from multispectral images and DSMs, then reconstructs polygons. All these methods rely on sufficient coverage representative points for each of the segmented building components in order to detect the components.

Voting-based algorithms bypass the need for a dedicated segmentation step by allowing individual points or seeds in a point cloud to vote for parameters of the structures to which they belong (Qi et al., 2019). Notable examples include PPGNet (Zhang et al., 2019) for 2D line and vertex extraction, and its 3D adaptation in Point2Roof (Li et al., 2022). One limitation is that it produces wireframes that may not form valid, watertight 3D models. In (Liu et al., 2024), the authors use a transformer-based architecture to detect vertices and then deduce faces from them. This produces a set of polygons, which do not necessarily form a valid polyhedron.

Transformers leverage self-attention mechanisms to capture complex dependencies within the data (Vaswani et al., 2017).

By using explicit information about feature locations, these models are particularly effective in environments where understanding of spatial relationships within data is necessary. Originally designed for natural language processing, they have been extended to various domains, including 3D data analysis and point cloud processing (Lu et al., 2022). DETR (Detection Transformers) approaches solve object detection as a direct set prediction problem (Carion et al., 2020). The architecture comprises a feature extraction step followed by a transformer that uses positional embeddings. Unlike voting-based methods, DETR approaches eliminate the need for non-maximum suppression or grouping and provide a more flexible output format, such as polygons. This has been shown in 2D building footprint prediction, as seen in PolyBuilding (Hu et al., 2022), and adapted to 3D contexts for object detection in works such as 3DETR presented by (Misra et al., 2021).

3. Materials and Methods

We frame building reconstruction as an object detection problem, where the objects are building facets represented by planar polygons contained within half-planes. Each half-plane is defined by its centroid point $\mathbf{p}$ and the normal vector $\mathbf{n}$ pointing outward. In addition, each plane is labelled with a class label to indicate whether it is a floor, facade, or roof plane. The building model can then be reconstructed using Boolean intersections of the half-planes, producing a convex, watertight volume.
We adapt the Detection Transformer (DETR) architecture (Carion et al., 2020) for this task (see Fig. 2), which originally performs bounding box predictions from images. An advantage of the DETR architectures and transformers in general is their independence of the data format. While convolutional neural networks (CNNs) encode the grid structure of the input images into the architecture, transformers only rely on the positional encoding of the input features.

3.1 Architecture

To solve the object detection task, we adapt the original DETR architecture (Carion et al., 2020) from detecting 2D bounding boxes to 3D plane primitives. We formulate the task as set prediction and adapt the DETR architecture to directly predict the set of planes and their parameters. As backbone for point clouds, we use a simple PointNet++(Qi et al., 2017) network with two layers. In this step, the input point cloud is downsampled and aggregated from N to N' coordinates and point features, which are subsequently projected to the transformer dimensions. We use the original DETR transformer proposed by Carion et al. (Carion et al., 2020), which was pretrained on 2D object detection. In contrast to 3DETR (Misra et al., 2021), we use learnt parametric queries as also used in the original DETR and do not use the input point cloud to create queries.

3.2 Positional Encoding

Transformers require positional information to process inputs effectively. Following Vaswani et al. (Vaswani et al., 2017), we employ **sine-based positional encoding**, which uses sinusoidal functions to encode spatial positions. Although designed for word order in text models, it also works for multidimensional coordinates. Their performance is empirically evaluated in section 4.1 against other encoding methods such as Fourier-based.

3.3 Heads and Losses

The planes are represented as $\{p_i\} = (x, n, c)$, where x is the centroid, n is the normal vector, and c is the class. The model consists of two heads. The first is a classification head to predict the plane type (floor, facade, roof and the DETR-specific 'no-object' class) using cross-entropy loss

$$L_{\text{CE}}(c, \hat{c}) = -\sum_i c_i \log(\hat{c}_i).$$

Furthermore, we use a plane regression head that outputs three parameters for the centroid of the plane, as well as three parameters for the normal. The composite loss function comprises:

- Huber loss for the six-dimensional plane parameters

$$L_{\text{planes},\delta}(p, \hat{p}) = \begin{cases} \frac{1}{2}(p-\hat{p})^2 & if\ |(p-\hat{p})| < \delta \\ \delta((p-\hat{p}) - \frac{1}{2}\delta) & otherwise \end{cases}$$

- Cosine similarity loss for the angle discrepancy between the predicted and ground truth normals

$$L_{\cos}(n, \hat{n}) = 1 - \frac{\mathbf{n} \cdot \hat{\mathbf{n}}}{||\mathbf{n}||_2 ||\hat{\mathbf{n}}||_2}$$

- The Euclidean distance from the predicted centroid to the actual plane

$$L_{\text{dist}}((x, n), \hat{x}) = |(x - \hat{x}) \cdot n|$$

The final loss is thus a weighted sum

$$loss = w_1 L_{\text{CE}} + w_2 L_{\text{planes}} + w_3 L_{\cos} + w_4 L_{\text{dist}}.$$

3.4 Bipartite Matching

As the DETR architecture predicts the set of planes in arbitrary order, it is necessary to match its outputs with the ground truth planes. To solve this optimisation problem, we use the Hungarian matching algorithm. The cost function is a weighted sum of the l_1-norm between the plane parameters and the class error, defined as $1 - \hat{p}(c)$.

3.5 Training Dataset

We used the RoofN3D dataset from (Wichmann et al., 2018), which consists of rectangular buildings and provides both the point clouds and parameters of the building planes to reconstruct the 3D model. Although these point clouds are originally constructed using cadastral footprints, our approach does not use this additional information. From the RoofN3D dataset, we extracted all buildings with rectangular footprints resulting in 118.073 buildings. From these we reserved 20% as holdout data for testing.

3.6 Evaluation Metrics

We evaluated the model on the test set from RoofN3D using multiple performance metrics. To assess Intersection-over-Union (IoU), we converted the half-space representation into explicit mesh reconstructions using PyMesh (Zhou, 2020). While this step enables direct geometric comparison, it is not

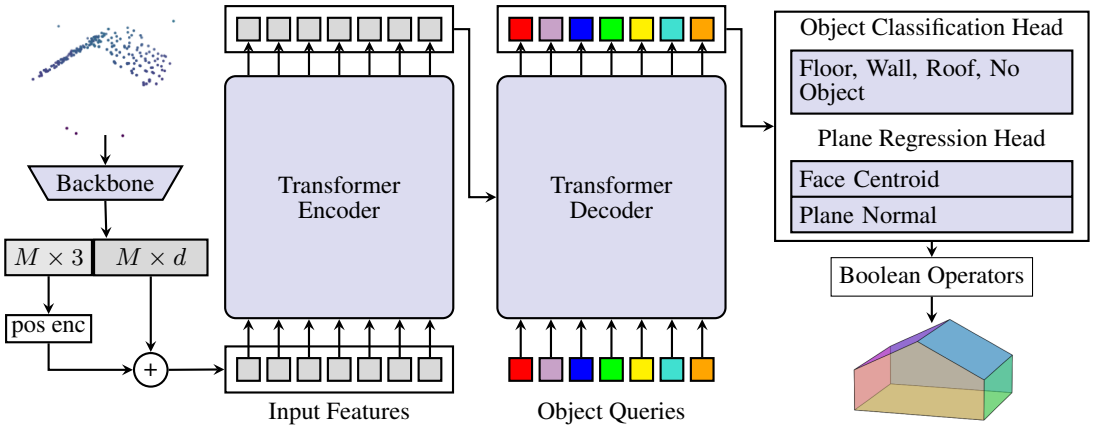

Figure 2. DETR-based architecture for detecting building faces from 3D point clouds.

always necessary, as a city model could also be stored in half-space form depending on the application. Further metrics are angle error, representative points distance, point-to-plane distance, and cardinal error. The angle error measures the angular discrepancy between predicted and actual normals. Representative points distance quantifies the Euclidean distance between the predicted and actual building face centroids. The point-to-plane distance assesses the proximity of the estimated point to the actual plane. Cardinal error captures discrepancies in the number of building faces

3.7 Additional Real World Testing Dataset

In order to test the model's generalisation capabilities, we applied the trained model to a subset of buildings from the AHN3 data in the City3D dataset presented in (Peters et al., 2022). To match the training dataset, we filtered the dataset to include only buildings whose footprints are roughly rectangular, defined by an overlap of at least 80% between their alpha shape and oriented bounding box when projected onto a 2D plane.

4. Results

The sample predictions of the generated 3D building models are shown in Figure 3. Typical model errors are building models that are more symmetric than the actual building. Our model achieved robust performance across various evaluation metrics. With a mean angle error of 1.7°, a mean representative points distance of 0.42 metres, and a mean point-to-plane distance of 0.16 meters, the model effectively captures the geometric characteristics of the buildings. The cardinal error was observed to be 0.08, indicating that the model estimates the correct number of faces in the building structures in most cases. Furthermore, a high IoU score of 0.88 was achieved. We found that large IoU errors often stem from data ambiguities, such as when distinguishing between integral building structures and adjacent elements like awnings is challenging. Angle discrepancies and slight plane translations are punished less harshly by the IoU. Notably, the generated buildings largely exhibited symmetrical structures and right angles, aligning well with typical architectural forms.

4.1 Impact of Positional Encodings

To incorporate positional information into the transformer architecture, we investigated three different positional encoding schemes in addition to the sine-based encoding proposed in 3.2: **Fourier-based**, **learned**, and **zero encoding**. Following the approach presented in (Wang et al., 2022), **Fourier-based encoding** employs a Fourier series to encapsulate positional information, which proves to be effective in capturing complex spatial relationships. In the case of a **learned encoding**, the positional encodings are initially randomised and subsequently updated by backpropagation, similar to the method described by (Yu et al., 2022). Although powerful, this approach lacks the ability to generalise to unseen positions, limiting its use in certain contexts. Lastly, **zero encoding** does not explicitly provide any positional information to the model, serving as a baseline to assess the effectiveness of the other encoding strategies. In Table 1, results for four types of position encodings are shown. We used sine-based, learnt, Fourier-based, and no positional encoding (referred to as ”zero” encoding). We only found small differences in the use of different positional encodings. This aligns with the findings of Misra et al. (Misra et al., 2021), who argued that the input features of the transformer encoder inherently contain coordinate information from the point cloud, making positional encoding less critical. Unlike text or images, point clouds are unordered by nature, making the transformer architecture particularly suitable. We observed no necessity for providing the decoder with a specific positional encoding. We hypothesize that this may be attributed to the relatively simple nature of the output space of our building dataset compared to the dataset used by Misra et al., which allows for the effective learning of queries even in a 3D context. In our case, the bottleneck seems to be the features and coordinates forwarded by the backbone or ambiguities in the training data itself rather than the positional encoding.

Positional Encoding	Learned	Fourier	Sine	Zero
Angle Error	1.7°	1.8°	$\mathbf{1.7^\circ}$	1.9°
Rep. Points Dist. [m]	0.43	0.46	**0.42**	0.45
Point-to-Plane Dist. [m]	0.16	0.16	**0.16**	0.18
Cardinal Error	0.08	**0.06**	0.08	0.06
IoU	0.88	0.88	0.88	**0.87**

Table 1. Evaluation of various position encoding methods.

4.2 Impact of the Training Data Distribution

As the training data plays an important role for machine learning models, we also investigated the errors in relation to the distribution of the training data. Fig. 5 shows that the error increases for planes that have fewer similar samples in the training dataset.

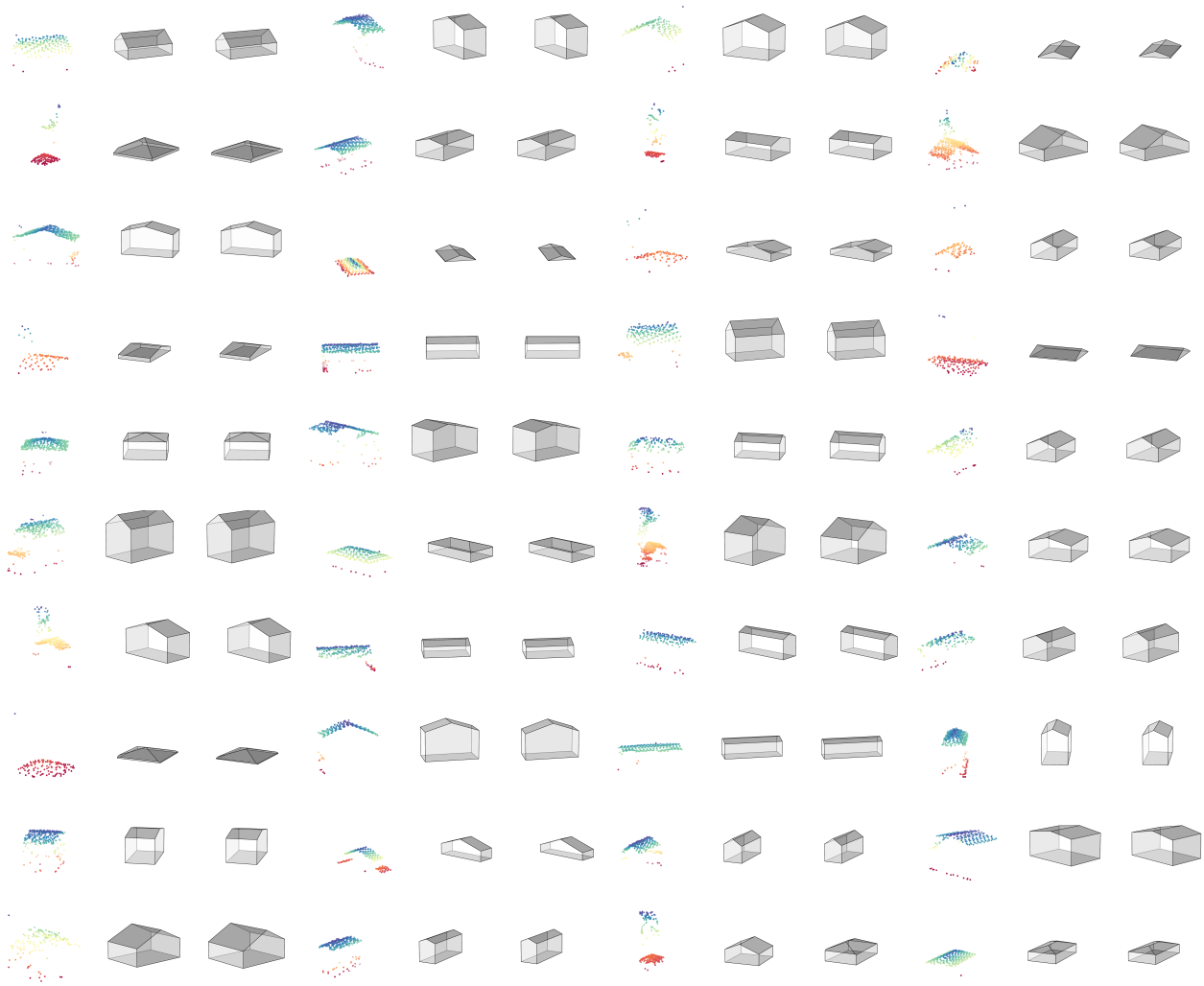

Figure 3. Predictions on an uncurated subset of the test set in groups of three: input point cloud, ground truth model, predicted model. The height of the points in the point cloud is colour-coded. Not that the scale of the point clouds does not match the scale of the 3D models.

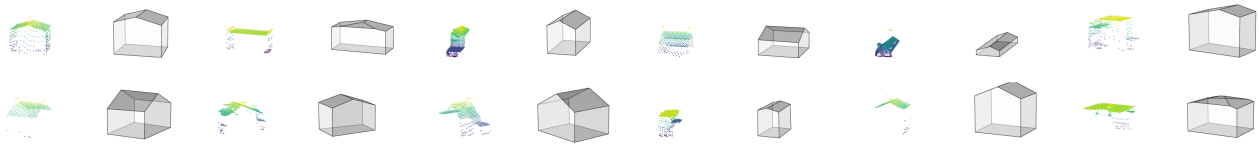

Figure 4. Alternatingly input point cloud from City3D dataset (AHN) and prediction

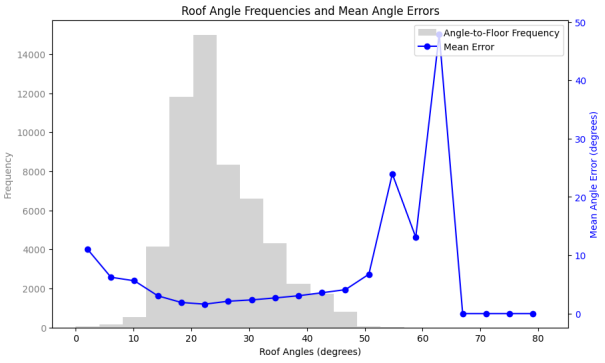

Figure 5. Histogram of the frequency of angles of the roof planes and median errors for the respective planes in the test dataset.

4.3 Real-World Test

Sample results for randomly selected point clouds of roughly rectangular buildings from the AHN3 part of the City3D dataset are shown in Figure 4. While all extracted models form correct, watertight 3D volumes and visually convincing buildings, their correspondence to the observed point cloud varies. As flat roofs were not part of the training data set, the model had to approximate them with multiple planar facets. This highlights its sensitivity to underrepresented architectural variations, as observed in Section 4.2, while also demonstrating its robustness in adapting to unseen structures.

5. Discussion

The adaptation of the DETR architecture from 2D to 3D introduces new challenges, particularly in how the model handles positional encodings, query initialization, and spatial feature representation. The results of our study indicate that the positional embedding used in the transformer model might not be as crucial as initially expected. A possible explanation is that the feature embeddings derived from the point cloud coordinates inherently carry sufficient spatial information, which reduces the need for explicit positional embeddings. Additionally, the downsampling operations performed in the point cloud backbone tend to degrade spatial precision, further minimising the impact of positional encoding, which is only based on the subsampled points. This suggests that point cloud processing pipelines may be optimised by focussing less on positional embeddings and more on robust feature extraction techniques. Queries in DETR-based architectures can be parametric (learned embeddings, as in DETR) or derived from input data (as in 3DETR, which samples points from the point cloud). Misra et al. argue that using points from the point cloud as queries improves the results, as 3D scenes are often too complex for parametric queries to generalise effectively (Misra et al., 2021). However, in our case, the situation differs: queries can learn a structured space of plausible buildings, rather than having to adapt to arbitrary 3D environments. In our controlled experiments with simple convex buildings, this approach is sufficient, as the learnt queries implicitly capture the regularities and constraints of the training distribution. For more complex structures, however, this remains an open question and future work should investigate whether incorporating queries derived from the point cloud improves reconstruction accuracy in more diverse architectural settings.

A major challenge for machine learning-based methods is the availability of large-scale high-quality datasets. Unlike traditional approaches, deep learning models require substantial training data, which is particularly difficult to obtain for 3D building reconstruction, as data sets must capture not only exterior surfaces but also structural relationships between building components. Ideally, such datasets should represent buildings in a way similar to Constructive Solid Geometry (CSG) trees, where each component is modelled as a solid. Many existing datasets, such as used for the ISPRS benchmark (Rottensteiner et al., 2014), lack the necessary structural detail and require significant preprocessing. Although formats like CityGML provide hierarchical representations of buildings, they do not explicitly define how individual planes or building parts relate. The *.obj format, commonly used for geometry storage, represents buildings as surface meshes rather than solid structures, making it unsuitable for operations that rely on Boolean intersections. In contrast, STEP and CAD formats inherently support solid representations, ensuring that each component is defined as a watertight polyhedron, preserving geometric integrity.

Another critical issue is the presence of biases in the model and the challenges in understanding what drives its behaviour. The proposed method has several advantages in meeting the outlined requirements: the geometries are always valid due to construction using Boolean operators, making them robust in terms of geometric integrity. The models are easy to edit as they contain relatively few primitives, which can be adapted by moving representative points and rotating planes. The resulting models are generally as simple as those provided in the training data. Aesthetically, the output is similar to the training dataset, and our results showed that the model can learn to favour symmetric constructions. The accuracy of the model is generally good, but may not always match the precision achievable by directly computing angles and other geometric details from the point cloud. Improving spatial accuracy could be achieved by using more sophisticated backbones or a deformable DETR model. However, it is important to note that biases may still arise, as the model may prioritise learnt higher-level assumptions, such as symmetry, over exact geometric details, especially in ambiguous cases. This opacity becomes particularly problematic when reconstructing buildings with uncommon architecture or when dealing with input point clouds that are of poor quality, e.g. those with occluded surfaces or high levels of noise. Unlike traditional RANSAC-based methods, where primitives are extracted based on the number of points supporting each geometric element, deep learning models do not offer such explicit interpretability, making the output less predictable under varied conditions. On the other hand, it can be argued that human modelers are also not fully transparent in the way they extract the 3D buildings, and the deep learning method tries to mimic their behaviour as closely as possible.

6. Conclusion

In this work, we have adapted the DETR architecture to building face detection from 3D point clouds. From these, watertight 3D building models can be inferred using half-space modelling and Boolean operations. We showed that the model is capable of learning to also predict building facets that do not have representative points, which is a common issue of ALS point clouds, which might not capture all parts of the buildings due to scanning direction and occlusions.

This is performed by the model by learning higher-level geometric rules, such as favouring 90°-degree angles and sym-

metry. This makes it adaptable to many architectural styles without providing explicit model rules. The roof archetypes used in (Henn et al., 2013) can thus all be learnt without explicitly specifying them. The downside of the model is the requirement for a large amount of diverse training data, which it shares with most data-driven methods. Furthermore, it can happen that the network proposes a building model which does not closely relate to the given input data, but looks convincing. To address the challenge of acquiring training data, we propose the workflow of initially creating 3D models using a fully automatic method that does not rely on machine-learning and then correct them to obtain a training dataset. If footprints are not available, input point clouds could be generated through semantic segmentation of ALS point clouds using existing models such as RandlaNet (Hu et al., 2020) or KPConv (Thomas et al., 2019), followed by an extraction of individual buildings. An approach suitable for creating complex CSG trees is presented in (Li and Shan, 2022). So far, we focused on simple buildings with rectangular footprints, which are available in the RoofN3D dataset. It would be directly applicable to all datasets containing convex 3D models but would require further adaptions for non-convex buildings. This limitation stems from the fact that intersections of half-planes are always convex. Extensions to more complex buildings with non-convex geometries would require training data that include information of the whole CSG tree; thus datasets such as Vaihingen from the ISPRS benchmark (Rottensteiner et al., 2014) which only include the facets of the boundary representation are not suitable. Future work will focus on creating datasets to accommodate complex building structures.

References

Ballard, D. H., 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 111–122.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., Çöltekin, A., 2015. Applications of 3D City Models: State of the Art Review. *ISPRS International Journal of Geo-Information*, 4(4), 2842–2889.

Bizjak, M., Žalik, B., Lukač, N., 2021. Parameter-Free Half-Spaces Based 3D Building Reconstruction Using Ground and Segmented Building Points from Airborne LiDAR Data with 2D Outlines. *Remote Sensing*, 13(21), 4430.

Buyukdemircioglu, M., Kocaman, S., Kada, M., 2022. Deep Learning for 3d Building Reconstruction: A Review. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2022, 359–366.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S., 2020. End-to-end object detection with transformers. *European conference on computer vision*, Springer, 213–229.

Carpentieri, M., Robins, A. G., 2015. Influence of urban morphology on air flow over building arrays. *Journal of Wind Engineering and Industrial Aerodynamics*, 145, 61–74.

Gao, X., Shen, S., Zhou, Y., Cui, H., Zhu, L., Hu, Z., 2018. Ancient Chinese architecture 3D preservation by merging ground and aerial point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143, 72–84.

Haala, N., Kada, M., 2010. An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6), 570–580.

Henn, A., Gröger, G., Stroh, V., Plümer, L., 2013. Model driven reconstruction of roofs from sparse LIDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76, 17–29.

Hu, P., Miao, Y., Hou, M., 2021. Reconstruction of complex roof semantic structures from 3D point clouds using local convexity and consistency. *Remote Sensing*, 13(10), 1946.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11108–11117.

Hu, Y., Wang, Z., Huang, Z., Liu, Y., 2022. PolyBuilding: Polygon Transformer for End-to-End Building Extraction. arXiv:2211.01589 [cs].

Kada, M., 2022. 3d Reconstruction of Simple Buildings from Point Clouds Using Neural Networks with Continuous Convolutions (convpoint). *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-4/W4-2022, 61–66.

Li, H., Xiong, S., Men, C., Liu, Y., 2023. Roof Reconstruction of Aerial Point Cloud Based on BPPM Plane Segmentation and Energy Optimization. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16, 5828–5848. Conference Name: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

Li, L., Li, Q., Xu, G., Zhou, P., Tu, J., Li, J., Li, M., Yao, J., 2024. A boundary-aware point clustering approach in Euclidean and embedding spaces for roof plane segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 218, 518–530.

Li, L., Song, N., Sun, F., Liu, X., Wang, R., Yao, J., Cao, S., 2022. Point2Roof: End-to-end 3D building roof modeling from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 193, 17–28.

Li, Z., Shan, J., 2022. RANSAC-based multi primitive building reconstruction from 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 185, 247–260.

Liu, K., Ma, H., Zhang, L., Liang, X., Chen, D., Liu, Y., 2023. Roof Segmentation From Airborne LiDAR Using Octree-Based Hybrid Region Growing and Boundary Neighborhood Verification Voting. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 16, 2134–2146.

Liu, Y., Obukhov, A., Wegner, J. D., Schindler, K., 2024. Point2Building: Reconstructing buildings from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 215, 351–368.

Lu, D., Xie, Q., Wei, M., Gao, K., Xu, L., Li, J., 2022. Transformers in 3D Point Clouds: A Survey. arXiv:2205.07417 [cs].

Misra, I., Girdhar, R., Joulin, A., 2021. An End-to-End Transformer Model for 3D Object Detection. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Montreal, QC, Canada, 2886–2897.

Oude Elberink, S., Vosselman, G., 2011. Quality analysis on 3D building models reconstructed from airborne laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2), 157–165.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., Stoter, J., 2022. Automated 3D Reconstruction of LoD2 and LoD1 Models for All 10 Million Buildings of the Netherlands. *Photogrammetric Engineering & Remote Sensing*, 88(3), 165–170.

Qi, C. R., Litany, O., He, K., Guibas, L., 2019. Deep Hough Voting for 3D Object Detection in Point Clouds. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Seoul, Korea (South), 9276–9285.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Romero Rodríguez, L., Duminil, E., Sánchez Ramos, J., Eicker, U., 2017. Assessment of the photovoltaic potential at urban level based on 3D city models: A case study and new methodological approach. *Solar Energy*, 146, 264–275.

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J. D., Breitkopf, U., Jung, J., 2014. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 256–271.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for Point-Cloud Shape Detection. *Computer Graphics Forum*, 26(2), 214–226.

Soleimani Vostikolaei, F., Jabari, S., 2023. Large-Scale LoD2 Building Modeling using Deep Multimodal Feature Fusion. *Canadian Journal of Remote Sensing*, 49(1), 2236243.

Sun, X., Guo, B., Li, C., Sun, N., Wang, Y., Yao, Y., 2024. Semantic Segmentation and Roof Reconstruction of Urban Buildings Based on LiDAR Point Clouds. *ISPRS International Journal of Geo-Information*, 13(1), 19. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L. J., 2019. Kpconv: Flexible and deformable convolution for point clouds. *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Tian, Y., Song, W., Chen, L., Sung, Y., Kwak, J., Sun, S., 2020. Fast Planar Detection System Using a GPU-Based 3D Hough Transform for LiDAR Point Clouds. *Applied Sciences*, 10(5), 1744. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

Tomljenovic, I., Höfle, B., Tiede, D., Blaschke, T., 2015. Building Extraction from Airborne Laser Scanning Data: An Analysis of the State of the Art. *Remote Sensing*, 7(4), 3826–3862. Number: 4 Publisher: Multidisciplinary Digital Publishing Institute.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vosselman, G., Dijkman, S., 2001. 3D building model reconstruction from point clouds and ground plans. *International archives of photogrammetry remote sensing and spatial information sciences*, 34(3/W4), 37–44.

Wang, Y., Guizilini, V. C., Zhang, T., Wang, Y., Zhao, H., Solomon, J., 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. *Conference on Robot Learning*, PMLR, 180–191.

Wichmann, A., Agoub, A., Kada, M., 2018. Roofn3d: Deep Learning Training Data for 3d Building Reconstruction. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2, Copernicus GmbH, 1191–1198. ISSN: 1682-1750.

Xu, Y., Jubanski, J., Bittner, K., Siegert, F., 2024. Roof plane parsing towards LoD-2.2 building reconstruction based on joint learning using remote sensing images. *International Journal of Applied Earth Observation and Geoinformation*, 133, 104096.

Yu, F., Chen, Z., Li, M., Sanghi, A., Shayani, H., Mahdavi-Amiri, A., Zhang, H., 2021. CAPRI-Net: Learning Compact CAD Shapes with Adaptive Primitive Assembly. arXiv:2104.05652 [cs].

Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J., 2022. Point-BERT: Pre-training 3D Point Cloud Transformers with Masked Point Modeling. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, New Orleans, LA, USA, 19291–19300.

Zhang, Z., Li, Z., Bi, N., Zheng, J., Wang, J., Huang, K., Luo, W., Xu, Y., Gao, S., 2019. Ppgnet: Learning point-pair graph for line segment detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7105–7114.

Zhou, Q., 2020. PyMesh: Geometry Processing Library for Python. Accessed on 20 August 2024. `https://pymesh.readthedocs.io/en/latest/`.