A METHOD FOR ROOF WIREFRAME RECONSTRUCTION BASED ON SELF-SUPERVISED PRETRAINING

Hongxin Yang[1], Shangfeng Huang[1], and Ruisheng Wang[1, 2, *]

[1]Department of Geomatic Engineering, Schulich School of Engineering, University of Calgary, Canada , - { hongxin.yang, shangfeng.huang, ruiswang}@ucalgary.ca
[2]School of Architecture and Urban Planning, Shenzhen University, China.

KEY WORDS: self-supervised learning (SSL), wireframe reconstruction, edge parameter regression, edge-based non-maximum suppression

ABSTRACT:

In this paper, we present a two-stage method for roof wireframe reconstruction employing a self-supervised pretraining technique. The initial stage utilizes a multi-scale mask autoencoder to generate point-wise features. The subsequent stage involves three steps for edge parameter regression. Firstly, the initial edge directions are generated under the guidance of edge point identification. The next step employs edge parameter regression and matching modules to extract the parameters (namely, direction vector and length) of edge representation from the obtained edge features. Finally, a specifically designed edge non-maximum suppression and an edge similarity loss function are employed to optimize the representation of the final wireframe models and eliminate redundant edges. Experimental results indicate that the pre-trained self-supervised model, enriched by the roof wireframe reconstruction task, demonstrates superior performance on both the publicly available Building3D dataset and its post-processed iterations, specifically the Dense dataset, outperforming even traditional methods.

1. INTRODUCTION

Accurate 3D roof models play a crucial role in architectural applications and urban planning. Existing 3D roof reconstruction methods are mainly divided into two categories: image-based and point cloud-based (Wang and Zakhor, 2022). In this paper, we focus on reviewing research related to point cloud-based reconstruction. In comparison to the 3D roof mesh representation (constructed by vertices, lines, and faces) in surface reconstruction, a wireframe model (constructed by vertices and lines) is a simplified representation of a complex 3D shape. It can be generated from a set of point cloud data and can also serve as a foundation for constructing mesh models. Due to the characteristics of wireframe model representation, constructed by vertices and lines, many researchers have transformed the problem of wireframe reconstruction into vertices identification and edge linking problems.

With the widespread adoption of deep learning techniques and considering that traditional methods may introduce accumulated errors, some 3D wireframe reconstruction methods based on end-to-end approaches have been proposed. A pioneering work is Point2Roof (Li et al., 2022), which represents the first end-to-end 3D roof modeling from airborne LiDAR point clouds. It employs PointNet++ (Qi et al., 2017b) as a backbone to extract point-wise features and identifies a series of candidate corner points. These candidate corner points are clustered into a set of initial vertices, and a paired point attention module is proposed to predict the final accurate vertices. However, this method is tested on artificial roof datasets that encompass a limited number of 16 simple roof types. Following this trajectory, Wang et al. (Wang et al., 2023) proposed a supervised-based baseline to predict the final wireframe model on an urban-scale dataset consisting of more than 160 thousand real-world buildings. WireframeNet (Cao et al., 2023) utilizes the medial axis transform technique to filter the original input point cloud. It integrates the corner and edge points’ information and analyzes the connectivity between the edge points to construct the complete wireframe structure. However, the aforementioned methods are supervised-based and require a large amount of labeled data. Given the expensive cost of manually labeling data, SSL is proposed to address this problem, wherein the model is trained with supervisory signals generated from the data itself. In previous work (Wang et al., 2023), a self-supervised baseline is proposed to identify corner points and use a graph neural network to generate the wireframe model. In essence, this method also follows the common route of corner identification and edge prediction, which may fail in the presence of missing corner points and sparsity in real-world datasets.

In this paper, we propose a pretrain-finetune pattern for 3D roof wireframe reconstruction, aiming to address the limitations encountered in real-world datasets. Specifically, we adopt a mask autoencoder as our self-supervised feature extractor. The input point clouds are partitioned into several patches using farthest point sampling and Ball Query techniques. These patches are categorized into visible and invisible patches, and the visible patches are fed into a mask autoencoder with a multi-scale feature mechanism to extract point-wise features. Subsequently, these point-wise features are input into a specifically designed edge point regression module to generate the initial direction of the edge. The generation of this initial edge direction is determined through an efficient 3D line fitting method. The output, including edge point identification results and edge initial direction, is then fed into the edge regression module to obtain the edge’s equation, which comprises the direction vector and length of the edge. More precisely, the edge parameter regression module is designed to parse the obtained edge features (including edge confidence scores, edge length, direction vector, and direction offset) by constructing multiple Multilayer Perceptron (MLPs) to regress distinct parameterized edges. Additionally, bipartite edge matching and edge non-maximum sup-

https://doi.org/10.5194/isprs-annals-X-2-2024-239-2024 | © Author(s) 2024. CC BY 4.0 License.

pression strategies are employed to generate accurate wireframe models.

Our main contributions can be summarized as follows:

(1) To the best of our knowledge, our proposed method is the first to utilize a pretrain-finetune strategy for edge regression under the guidance of edge point identification. The pre-trained self-supervised model, when combined with the downstream task of roof wireframe reconstruction, demonstrates superior performance.

(2) We introduce a novel method for parameterized edge extraction that leverages edge point information to establish the initial edge's direction vector. This initial direction vector, combined with edge direction, direction offset, and edge length, is used iteratively to generate wireframe drafts. Subsequently, edge-based non-maximum suppression is applied to eliminate redundant edges.

2. RELATED WORK

2.1 SSL for point clouds based on mask recovery

The fundamental concept underlying mask recovery in SSL methods involves acquiring point cloud representations through the reconstruction of corrupted point clouds, with the aim of recovering the original structure to the greatest extent possible. For example, Point-BERT (Yu et al., 2022) randomly masks a portion of the input point cloud and employs a BERT-style transformer to reconstruct the invisible tokens. This reconstruction is performed under the supervision of visible tokens obtained from a pre-trained discrete Variational Autoencoder (dVAE). However, this approach tends to overly depend on data augmentation. To address this issue, Point-MAE (Pang et al., 2022) introduces a unified framework of mask autoencoder. It utilizes the standard transformer as the backbone, featuring an asymmetric encoder-decoder architecture to effectively recover the masked data with a high ratio. In contrast, Occupancy-MAE (Min et al., 2023) is proposed to reconstruct masked voxels by relying only on a small number of visible voxels. This is achieved by combining a range-aware random masking strategy and incorporating a pretext task involving occupancy prediction techniques, particularly when the input point clouds are divided into multiple voxels. Continuing along this path, we employ the mask recovery-based autoencoder as our self-supervised feature extractor.

2.2 Wireframe reconstruction

The detection of corner points, akin to 3D key points, plays a crucial role in capturing and representing structural information. In previous work, USIP (Li and Lee, 2019) utilized a feature proposal network to obtain features for key points along with their corresponding transformation pairs. The generated key points underwent optimization by minimizing the distances between detected key points in pairs of point clouds. Li et al. (Li et al., 2022) employed point-wise binary classification and point clustering techniques to identify initial vertices. The PointNet backbone was then utilized to extract vertex features and perform offset regression. A subsequent refinement process was applied to precisely determine the locations of the initial vertices, resulting in the generation of accurate vertex representations. Jiang et al. (Jiang et al., 2023) introduced a double-flow structure for extracting both semantic and offset information. These informative features were then leveraged to identify high-quality interest regions. Deep estimators were subsequently employed to predict corner point proposals within each interest region, facilitating the accurate detection of corner points. It is noteworthy, however, that these methods belong to the category of supervised learning approaches, necessitating a substantial number of annotated point clouds. Furthermore, these methods have primarily been applied to synthetic datasets and regular geometric, close-range objects.

Additionally, alternative methods based on different patterns have been proposed. LC2WF (Yicheng Luo and Bao, 2022) introduced a Line-Patch transformer-based network to construct building wireframes by extracting junctions and connectivities from line cloud datasets. Ma et al. (Ma et al., 2022) presented a deep spatial gestalt model designed to infer the relationship between visible and invisible cues of 3D structure. Furthermore, some research has focused on achieving the goal of generating wireframes by extracting line segments. In particular, there are works dedicated to segmenting the input point cloud into a collection of facets to extract line segments from large-scale point clouds (Lin et al., 2017) (Lin et al., 2015). In contrast to these approaches, our proposed roof wireframe reconstruction method omits the corner detection steps and instead employs a regression approach for the edges. This is achieved under the guidance of the initial direction generation of the edge points.

3. OUR PROPOSED METHOD

3.1 Overview of experimental datasets

We present a comprehensive description of the publicly available Building3D dataset and its post-processed Dense dataset. We utilize this description to elucidate the motivation behind constructing a Dense dataset and introducing our proposed approach. As depicted in Fig.1(a), we randomly selected four roof datasets from the public Building3D dataset (Wang et al., 2023). However, this selection revealed certain issues, including noise and missing corner points, denoted by red-dotted circular shapes. The conventional wireframe reconstruction pattern of "corner detection + edge prediction" encountered challenges under these conditions. In response to this, we curated a Dense dataset, as illustrated in Fig.1(b), with the aim of observing the performance disparity of our proposed method on both dataset versions.

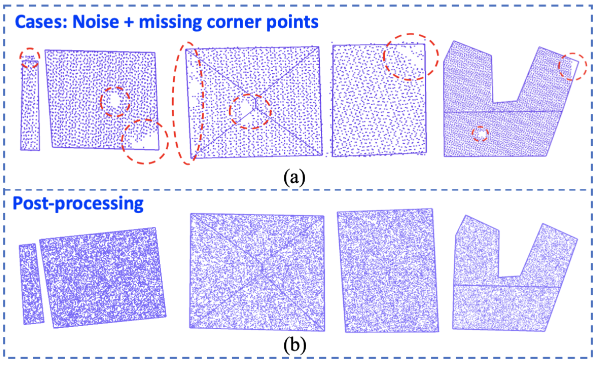

Figure 1. Characterization of experimental datasets: (a) illustration of the public Building3D dataset and (b) its post-processed Dense dataset.

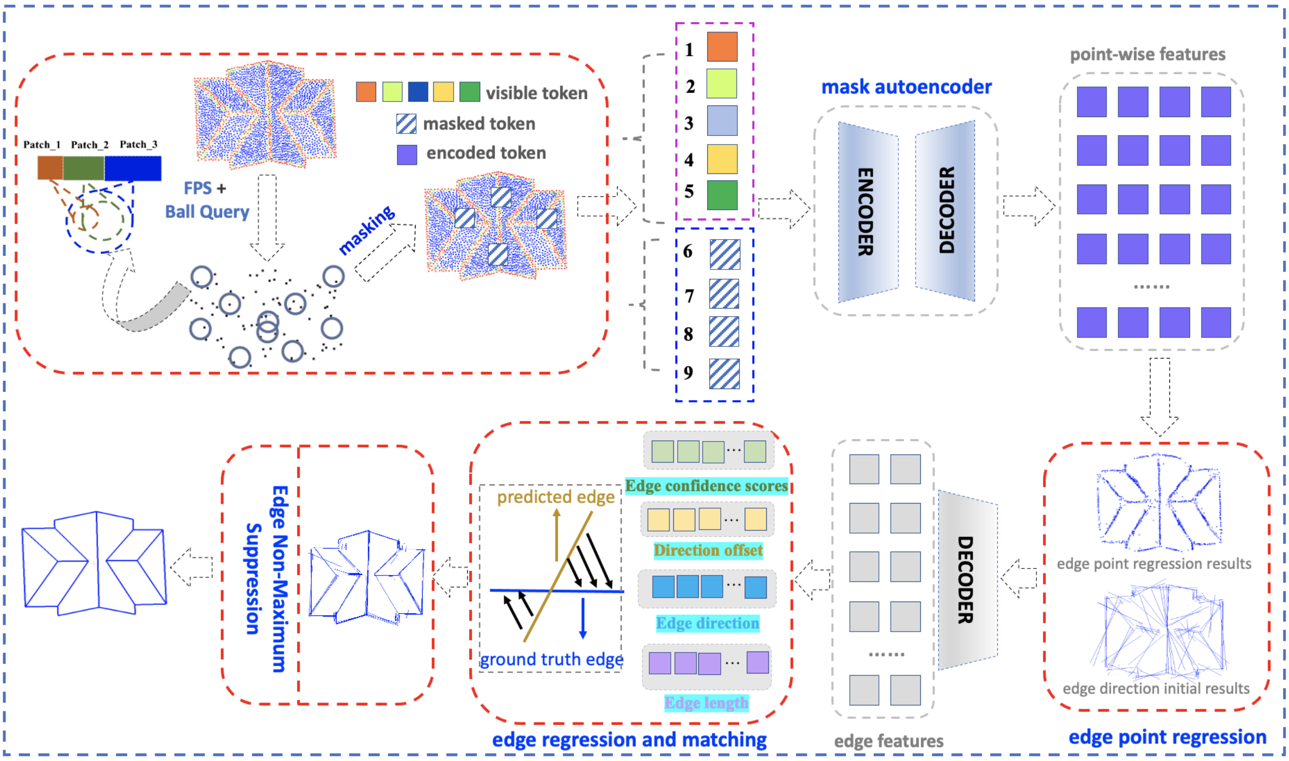

Figure 2. Workflow of our proposed methodology.

3.2 Workflow of our proposed methodology

Self-supervised feature extractor and edge point regression module: The Point-MAE (Pang et al., 2022) serves as a self-supervised feature extractor backbone, facilitating the extraction of multiple-scale features. For instance, as depicted in the upper-left corner of Fig.2, the grey points and circles represent the output derived from farthest point sampling (FPS) and Ball Query operations on the original input point cloud. The patches—**Patch**_1, **Patch**_2, and **Patch**_3—illustrate point sets at varying scales generated by applying the Ball Query operation to the original input. Subsequently, a random masking strategy is employed on the acquired point sets at different scales, segregating them into visible and invisible segments. The unmasked portions are then utilized as input to an encoder-decoder structure, facilitating the extraction of point-wise features. The point-wise features are fed into an edge point regression module to derive the identification results for the edge points. Subsequently, an efficient 3D line fitting strategy is applied to extract the edge features.

Edge regression and matching strategies: The purpose of the edge regression module is to extract the parameters defining edge representation from the acquired edge features. This module comprises four dedicated MLPs, each operating on specific tensors (Edge confidence scores, Direction offset, Edge direction, and Edge length) to regress their corresponding edge parameters. Initially, an Edge Confidence Scores MLP is employed to derive confidence scores for the edges. Subsequently, the orientation of the edge is decomposed into three components along the XYZ axis. This is achieved using a Direction Offset MLP to predict three component values on the XYZ axis. Finally, an Edge Length MLP is utilized to predict the residual offsets from candidate points to midpoints of the edges. Moreover, the conventional algorithm employed for solving Bipartite Edge Matching is the Hungarian algorithm (Zhang et al., 2016). Nevertheless, these methods prove unsuitable for our case, given the absence of a corner detection process in the 3D wireframe reconstruction. A novel edge similarity is formulated based on three distinct factors: the distance between the two lines, edge length, and edge direction.

*(1) **The distance between the two lines.*** The Hausdorff Edit Distance (HED) (Fischer et al., 2015) is employed for pairwise node matching. Each node in one graph is compared with every node in the other graph, akin to comparing subsets of a metric space using the Hausdorff distance. For example, we have defined two sets of edges, denoted as $\boldsymbol{A}$ and $\boldsymbol{B}$, where each edge in these sets is represented by $\boldsymbol{A}_i$ and $\boldsymbol{B}_j$. The points belonging to these edges are denoted as $\boldsymbol{A}_i$ and $\boldsymbol{B}_j$. The Hausdorff distance is defined with respect to the metric $dis(\boldsymbol{A}_i, \boldsymbol{B}_j)$ in the following manner:

$$\begin{aligned} H(\boldsymbol{A},\boldsymbol{B}) = max(&\max_{\boldsymbol{A}_i\in \boldsymbol{A}}\min_{\boldsymbol{B}_j\in \boldsymbol{B}} dis(\boldsymbol{A}_i,\boldsymbol{B}_j), \\ &\max_{\boldsymbol{B}_j\in \boldsymbol{B}}\min_{\boldsymbol{A}_i\in \boldsymbol{A}} dis(\boldsymbol{A}_i,\boldsymbol{B}_j)) \end{aligned} \tag{1}$$

(2) ***Edge length.*** The proximity between two edges is quantified by the ratio of the minimum distance (d_{min}) between the end points of the two edges to the length of the shorter segment. This is defined as follows:

$$pro(\boldsymbol{A}_i,\boldsymbol{B}_j) = 1 - min(1, \frac{d_{min}}{min(len_{\boldsymbol{A}_i}, len_{\boldsymbol{B}_j})}) \tag{2}$$

(3) ***Edge direction.*** Cosine similarity (Hoe et al., 2021) is employed to assess the directional similarity between two edges, and it is defined as follows:

$$sim(\boldsymbol{A}_i,\boldsymbol{B}_j) = \frac{<\boldsymbol{A}_i,\boldsymbol{B}_j>}{||\boldsymbol{A}_i||\cdot||\boldsymbol{B}_j||} \tag{3}$$

The edge similarity is expressed by combining Eqs. (1), (2), and (3), and it is defined as follows:

$$\boldsymbol{Edges}_{sim} = \alpha \cdot H(\boldsymbol{A},\boldsymbol{B}) + \beta \cdot sim(\boldsymbol{A}_i,\boldsymbol{B}_j) + \gamma \cdot pro(\boldsymbol{A}_i,\boldsymbol{B}_j) \tag{4}$$

Here, α, β, and γ represent the balancing coefficients. In our experiment, these coefficients are set to $\alpha = 2$, $\beta = 0.5$, and $\gamma = 0.5$, respectively. The smaller the value in Eq. (4), the greater the similarity between the two edges.

Edge Non-Maximum Suppression: The obtained edge similarity representation and predicted edge confidence scores are amalgamated to function as inputs for eliminating redundant edges. Firstly, all edges are sorted in descending order based on confidence scores. Next, the edge similarity function is constructed, and a threshold is set to select similar edges. Thirdly, an empty list is initialized to store the selected candidate edges. Edges with confidence scores, starting from high to low, are iteratively added to the list if they meet the threshold criteria.

3.3 Loss function

The overall loss of the entire framework comprises three components: the self-supervised feature extractor module, edge point regression module, and edge regression and matching modules. **Self-supervised feature extractor loss:** It is computed using the Chamfer distance (Pang et al., 2022). **Edge point regression and edge length losses:** We present a illustrative example, as depicted in Fig.3, to elucidate our design of the edge point loss. Let any point in a point cloud be denoted as **p**; the distances between this point **p** and its neighboring edges (Line1, Line2, Line3, and Line4) are marked as d_1, d_2, d_3, and d_4, respectively. Assuming the distance between any point **p** and the projection point in Line1 is the shortest. In three-dimensional space, the points within the shortest distance are distributed on the surface of a sphere, such as a marked point termed as a Candidate point. Additionally, besides ensuring that the distance between any point **p** and the projection point is the shortest, there is a need to minimize the distance between a candidate point and the projection point. The L1-norm distance loss (He et al., 2017) is employed to enforce these constraints and optimize edge point regression and edge length. **Edge confidence scores and edge direction losses:** The optimization of predicted edge confidence scores and edge direction is achieved through the utilization of cross-entropy and cosine similarity loss functions, respectively.

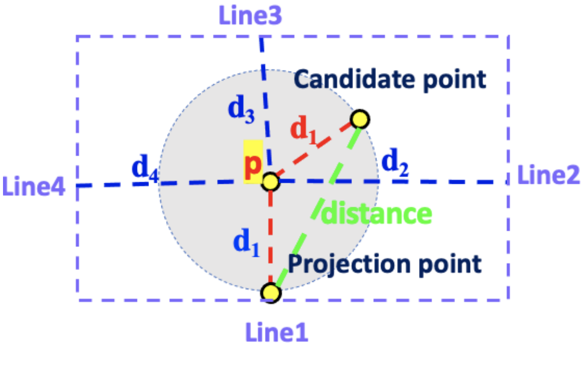

Figure 3. Illustration of the edge point regression loss.

Overall edge similarity loss: The edge similarity loss is computed by combining the aforementioned self-supervised loss, edge point loss, edge length, edge confidence scores, and edge direction loss with different multiplication factors, respectively.

4. EXPERIMENTS

4.1 Details of the experimental datasets

In this experimental section, the roof dataset utilized is primarily categorized into two versions: the Building3D dataset (Wang et al., 2023) and the Dense dataset. The generation of the Dense dataset was executed using CloudCompare software, facilitating the production of a dense point representation derived from the Building3D dataset. This was done to assess the performance difference of the proposed method on both Dense datasets without missing corner points and Building3D. The primary distinction between the Dense and Building3D datasets lies in the point density and the presence of missing corner points. The Dense dataset maintains uniform point density throughout, whereas the Building3D dataset may exhibit variations and lack some points. Specifically, the Dense and the Building3D datasets consist of 5,698 training point clouds and 583 testing point clouds.

4.2 Implementation details

During the SSL pipeline, we employ the AdamW optimizer with a learning rate set to 0.001 and train for 1000 epochs. The pre-training phase of our model consists of 300 epochs with a batch size of 128. In the fine-tuning process, the batch size is set to 2048 points. The point-wise features obtained from the SSL pipeline are then fed into the edge point identification module. These features, along with 128 query points and 64 query embeddings, are input into the decoder to obtain edge features with 256 channels. All code and models are trained using an RTX A6000 GPU with 48GB of memory.

4.3 Evaluation metric

The precision and recall of corners and edges are evaluated on the aforementioned three roof datasets. The metric is the same as (Wang et al., 2023), (Li et al., 2022). The average precision (AP) and recall (AR) for the predicted corners are calculated by considering the sets of predicted labels $\boldsymbol{G}$ and ground truth labels $\boldsymbol{Q}$. The AP and AR are formulated as follows:

$$\begin{gathered} AP = \frac{TP}{TP+FP}, \quad AR = \frac{TP}{TP+FN}, \\ F_1 = \frac{2*AP*AR}{AP+AR} \end{gathered} \tag{5}$$

where $TP \in |\boldsymbol{G} \cap \boldsymbol{Q}|$, $FP \in |\boldsymbol{G} \cap \overline{\boldsymbol{Q}}|$ and $FN \in |\overline{\boldsymbol{G}} \cap \boldsymbol{Q}|$. The symbols CP and EP, as well as CR and ER, have the same formulation as the previously mentioned AP and AR, respectively, and represent the precision and recall of corner points and edges. The symbols $| * |$ and $|\overline{*}|$ denote the cardinality of a given set and the complementary set of a given set. In addition, we also calculate the ACO which is the average offset between predicted corners and ground-truth corners.

4.4 Visual and comparative experiments

We demonstrate the iterative process of our proposed method using the Dense and Building3D datasets, as depicted in Fig.4 and Fig.5, respectively. In both Fig.4 and Fig.5, the first row

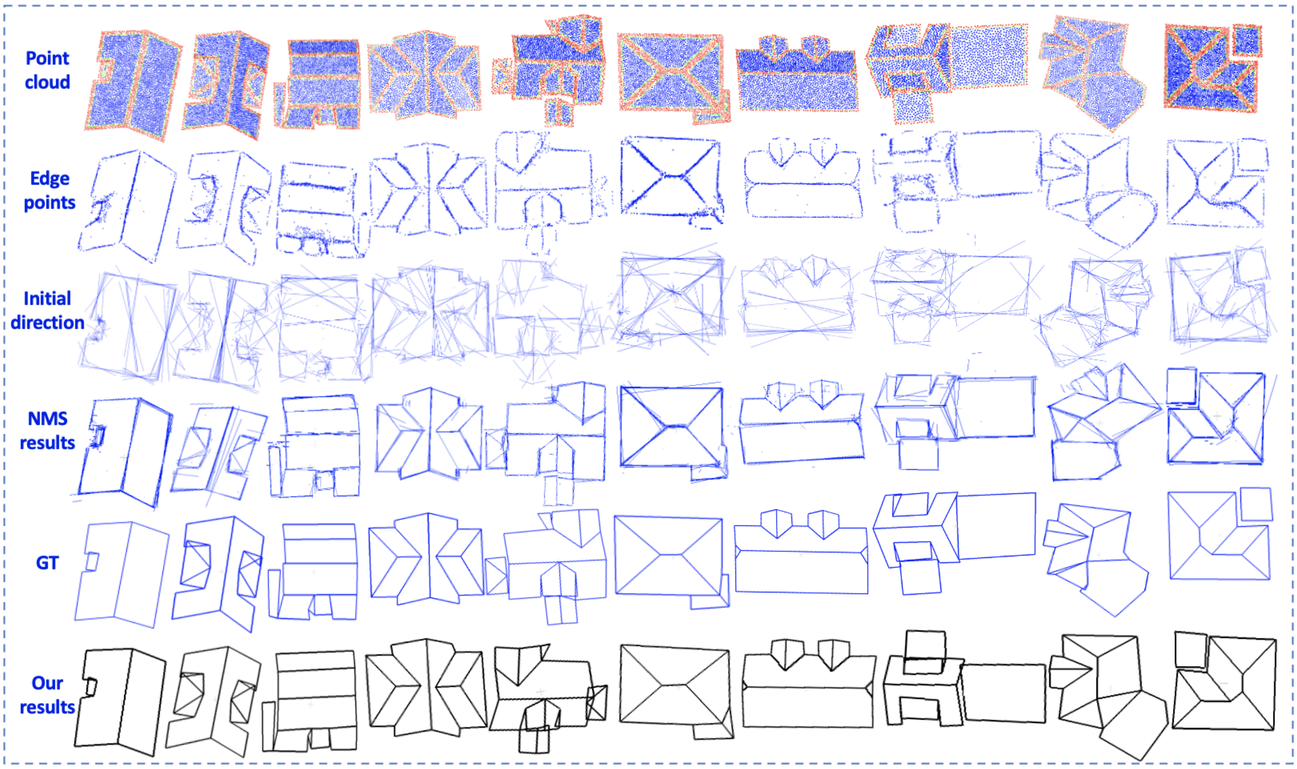

Figure 4. Results from the iterative step in our proposed method on the Dense dataset.

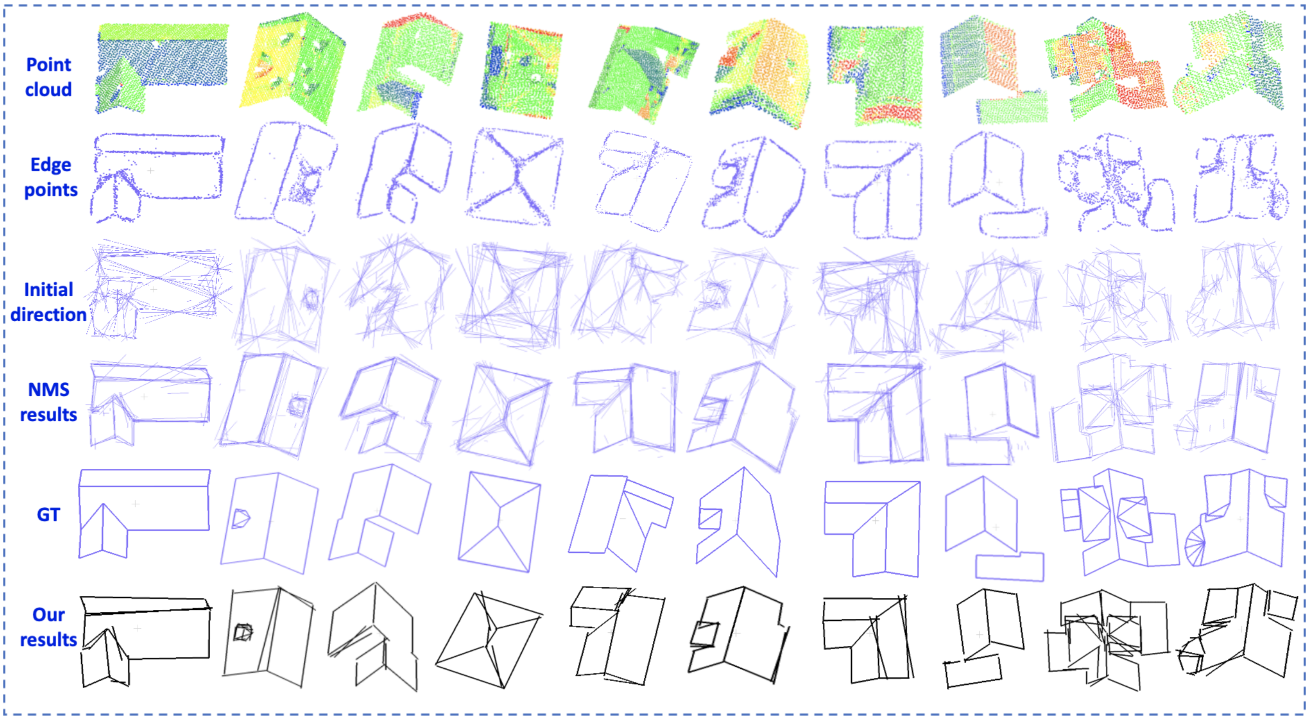

Figure 5. Results from the iterative step in our proposed method on the Building3D dataset.

Table 1. Comparative analysis of methodological performance on the Building3D Dataset.

Method (Building3D (Wang et al., 2023))		**ACO**	**CP//CR**	**F1**	**EP//ER**	**F1**
Supervised-based	PointNet* (Qi et al., 2017a)	0.36	0.71//0.50	0.59	0.81//0.26	0.39
	PointNet++* (Qi et al., 2017b)	0.34	0.79//0.52	0.63	0.84//0.33	0.47
	RandLA-Net* (Hu et al., 2020)	0.35	0.70//0.60	0.65	0.67//0.16	0.25
	DGCNN* (Phan et al., 2018)	0.32	0.73//0.58	0.65	0.81//0.30	0.44
	PAConv* (Xu et al., 2021)	0.33	0.75//0.57	0.65	0.85//0.31	0.45
	Stratified Transformer* (Lai et al., 2022)	0.38	0.72//0.51	0.62	0.75//0.22	0.34
	Point2Roof (Li et al., 2022)	0.30	0.66//0.48	0.56	0.71//0.26	0.38
	FG-Net* (Liu et al., 2022)	0.32	0.77//0.64	0.70	0.84//0.38	0.52
	Building3D (supervised) (Wang et al., 2023)	0.26	0.89//0.66	0.76	0.91//0.46	0.61
Self-supervised-based	Point-M2AE* (Zhang et al., 2022)	0.26	0.88//0.69	0.77	0.90//0.31	0.46
	Point-MAE* (Pang et al., 2022)	0.27	0.75//0.69	0.72	0.76//0.22	0.34
	Building3D (self-supervised) (Wang et al., 2023)	0.25	0.83//0.70	0.76	0.78//0.37	0.50
	Our proposed method (Dense)	**0.03**	**0.95//0.89**	**0.92**	**0.88//0.91**	**0.89**
	Our proposed method (Building3D)	**0.04**	**0.86//0.92**	**0.89**	**0.72//0.87**	**0.79**

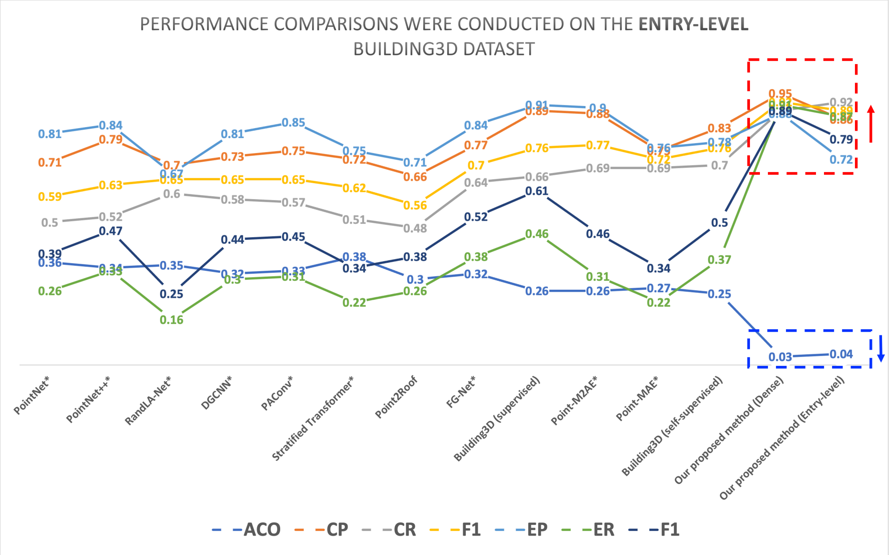

Figure 6. Comparative analysis of methodological performance on the Building3D Dataset.

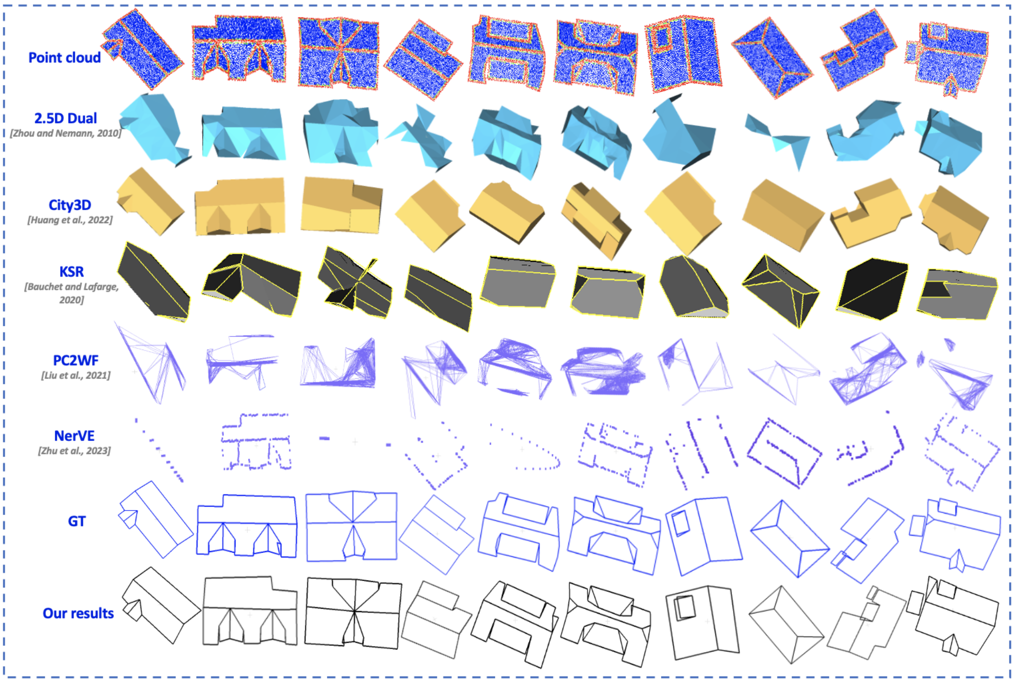

Figure 7. Qualitative comparisons among three conventional approaches, two affiliated deep learning methods, and our proposed method.

illustrates the original point cloud, while the second row displays the results of edge point identification. Proceeding to the third row, we provide an illustration depicting the generation of initial direction based on the outcomes of edge point detection. The fourth row presents the Non-Maximum Suppression (NMS) results processed through our proposed edge NMS method. The fifth and sixth rows represent the ground truth and the wireframe results of our reconstruction, respectively. We also evaluate the performance of our proposed method on the Building3D dataset using two different pretrained models on the Dense and Building3D datasets, comparing it with both non-transformer and transformer-based supervised learning, as well as other SSL approaches, as illustrated in Table.1 and Fig.6, respectively. The symbol ”*” denotes that the method is utilized as a feature extractor to extract point-wise features, followed by the corner detection and edge prediction modules for wireframe reconstruction, as reported in (Wang et al., 2023). The experimental results demonstrate that our proposed method on the Dense dataset exhibits a significant performance margin compared to the baseline provided by Building3D (Wang et al., 2023). The metrics CR and ER show increases of 19% and 54% (Building3D: 22% and 50%), respectively, with corresponding F1 scores showing improvements of 16% and 39% (Building3D: 13% and 29%) on the Building3D dataset. Furthermore, the ACO distance has significantly decreased. As illustrated in Fig.6, the red and blue dotted boxes indicate that higher values correspond to better performance in metrics such as CP, CR, F1 scores of corners, EP, ER, and F1 scores of edges, while lower values of ACO signify better results. These remarkable improvements can be attributed to the guidance provided by the edge point regression and edge initial directions in our proposed method.

Furthermore, we conduct visual comparisons with several experimental methods, including three traditional approaches: 2.5D Dual Contouring (Zhou and Neumann, 2010), City3D (Huang et al., 2022), and KSR (Bauchet and Lafarge, 2020). Additionally, we compare against two recent deep learning methods, PC2WF (Liu et al., 2021) and NerVE (Zhu et al., 2023), as illustrated in Fig.7. Compared with two deep learning-based methods, our reconstruction results achieve the most accurate representation of the roofs, as evidenced by the visualization results. Our reconstruction outputs are in the form of wireframes, which can be further processed by filling each face of the wireframe and assigning specific height values to generate the faces. This process enables us to convert the wireframes into mesh format. Such conversion provides a clear advantage when compared with the reconstruction results obtained from traditional methods. It is evident upon observation that the reconstruction results achieved by our proposed method surpass those of other compared methods, encompassing both traditional and related deep learning approaches.

5. CONCLUSION

This paper introduces a two-stage SSL method for the reconstruction of 3D roof wireframes. The proposed approach consists of four key components: a self-supervised feature extractor, an edge point regression module, edge regression and matching modules guided by edge initial direction, and an edge Non-Maximum Suppression module, aimed at eliminating redundant edges to achieve an accurate wireframe model. Notably, we incorporate an efficient edge point regression loss to identify the distribution of edge points and ensure the accuracy of the initial edge direction. Subsequently, parameterized edges resulting from this process undergo bipartite edge matching using the designed edge similarity algorithm. Experimental results demonstrate the superiority of our approach over state-of-the-art methods in terms of reconstruction performance on both Dense and Building3D datasets. Furthermore, the proposed method can be extended to other domains within architecture and urban planning where labeled data is limited, enabling the generation of more precise and diverse models.

REFERENCES

Bauchet, J.-P., Lafarge, F., 2020. Kinetic shape reconstruction. *ACM Transactions on Graphics (TOG)*, 39(5), 1–14.

Cao, L., Xu, Y., Guo, J., Liu, X., 2023. WireframeNet: A novel method for wireframe generation from point cloud. *Computers & Graphics*, 115, 226–235.

Fischer, A., Suen, C. Y., Frinken, V., Riesen, K., Bunke, H., 2015. Approximation of graph edit distance based on Hausdorff matching. *Pattern Recognition*, 48(2), 331–343.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. *Proceedings of the IEEE international conference on computer vision*, 2961–2969.

Hoe, J. T., Ng, K. W., Zhang, T., Chan, C. S., Song, Y.-Z., Xiang, T., 2021. One loss for all: Deep hashing with a single cosine similarity based learning objective. *Advances in Neural Information Processing Systems*, 34, 24286–24298.

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.

Huang, J., Stoter, J., Peters, R., Nan, L., 2022. City3D: Large-scale building reconstruction from airborne LiDAR point clouds. *Remote Sensing*, 14(9), 2254.

Jiang, T., Wang, Y., Zhang, Z., Liu, S., Dai, L., Yang, Y., Jin, X., Zeng, W., 2023. Extracting 3-D Structural Lines of Building from ALS Point Clouds using Graph Neural Network Embedded with Corner Information. *IEEE Transactions on Geoscience and Remote Sensing*.

Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J., 2022. Stratified transformer for 3d point cloud segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8500–8509.

Li, J., Lee, G. H., 2019. Usip: Unsupervised stable interest point detection from 3d point clouds. *Proceedings of the IEEE/CVF international conference on computer vision*, 361–370.

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.

Lin, Y., Wang, C., Chen, B., Zai, D., Li, J., 2017. Facet segmentation-based line segment extraction for large-scale point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 55(9), 4839–4854.

Lin, Y., Wang, C., Cheng, J., Chen, B., Jia, F., Chen, Z., Li, J., 2015. Line segment extraction for large scale unorganized point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 102, 172–183.

Liu, K., Gao, Z., Lin, F., Chen, B. M., 2022. Fg-net: A fast and accurate framework for large-scale lidar point cloud understanding. *IEEE Transactions on Cybernetics*, 53(1), 553–564.

Liu, Y., D'Aronco, S., Schindler, K., Wegner, J. D., 2021. Pc2wf: 3d wireframe reconstruction from raw point clouds. *arXiv preprint arXiv:2103.02766*.

Ma, W., Tan, B., Xue, N., Wu, T., Zheng, X., Xia, G.-S., 2022. How-3d: Holistic 3d wireframe perception from a single image. *2022 International Conference on 3D Vision (3DV)*, IEEE, 596–605.

Min, C., Xiao, L., Zhao, D., Nie, Y., Dai, B., 2023. Occupancy-MAE: Self-Supervised Pre-Training Large-Scale LiDAR Point Clouds With Masked Occupancy Autoencoders. *IEEE Transactions on Intelligent Vehicles*.

Pang, Y., Wang, W., Tay, F. E., Liu, W., Tian, Y., Yuan, L., 2022. Masked autoencoders for point cloud self-supervised learning. *European conference on computer vision*, Springer, 604–621.

Phan, A. V., Le Nguyen, M., Nguyen, Y. L. H., Bui, L. T., 2018. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, 108, 533–543.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

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

Wang, R., Huang, S., Yang, H., 2023. Building3d: A urban-scale dataset and benchmarks for learning roof structures from point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 20076–20086.

Wang, R., Zakhor, A., 2022. 3D Wireframe Reconstruction of Buildings from Point Cloud Data.

Xu, M., Ding, R., Zhao, H., Qi, X., 2021. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3173–3182.

Yicheng Luo, Jing Ren, X. Z. D. K. Y. X. P. W., Bao, L., 2022. Lc2wf:learning to construct 3d building wireframes from 3d line clouds. *Proceedings of the British Machine Vision Conference (BMVC)*.

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. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19313–19322.

Zhang, R., Guo, Z., Gao, P., Fang, R., Zhao, B., Wang, D., Qiao, Y., Li, H., 2022. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *Advances in neural information processing systems*, 35, 27061–27074.

Zhang, Z., Shi, Q., McAuley, J., Wei, W., Zhang, Y., Van Den Hengel, A., 2016. Pairwise matching through max-weight bipartite belief propagation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1202–1210.

Zhou, Q.-Y., Neumann, U., 2010. 2.5 d dual contouring: A robust approach to creating building models from aerial lidar point clouds. *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III 11*, Springer, 115–128.

Zhu, X., Du, D., Chen, W., Zhao, Z., Nie, Y., Han, X., 2023. Nerve: Neural volumetric edges for parametric curve extraction from point cloud. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13601–13610.