Research on BIM Reconstruction Method Using Semantic Segmentation Point Cloud Data Based on PointNet

Zhaoyang Xiong* and Ting Wang
Department of Civil Engineering and architecture, Nanchang Hangkong University, No.696, Fenghe South Avenue, Nanchang, Jiangxi Province, China
Email: shanyunfangmail@163.com

Abstract. As the construction industry is shifting from the construction of new buildings to the maintenance and use of existing buildings in recent years, the demand for automated building information models (BIM) creation is increasing. This paper uses the deep learning network PointNet to perform semantic segmentation on the public S3DIS point cloud data set, which means to assign the same type of point cloud building components in the data set to the same label, and the bounding box algorithm is been used to obtain the outer contour parameters of the segmented point cloud building components. Finally, the Dynamo, which is one of the Revit plug-ins, is used to perform parametric modeling according to the obtained parameters, and generates the BIM corresponding to the point cloud data set. The experimental results show that the method proposed in this paper can complete the parametric creation of BIM with high completeness based on the efficient segmentation of point clouds.

Keywords. Building Information Model, point cloud, deep learning, dynamo parametric automatic modeling.

1. Introduction
In the field of architecture, engineering, construction (AEC), semantically rich BIM are gaining more attention in the entire building’s life cycle, from design, construction phase to facility management phase [1]. For a large number of existing buildings, creating BIM for existing buildings can bring many benefits, such as providing service status information [2], maintenance cost budgeting [3], and energy analysis [4]. The three-dimensional point cloud is the most common type of three-dimensional data used to create BIM. The Pi corresponding to each point in the point cloud represents N dimensional vector which is composed of features such as the coordinate (x, y, z) of the point, normals and colors. The general practice is to convert the raw scanned scan point data into a high-level BIM representation, which is called the point-to-BIM transformation process [5]. Point cloud data needs to mark the measured value of each point in the point cloud when creating BIM, so that the points belonging to the category are given the same label. In other words, the point cloud is segmented at the semantic information level, that is, point cloud semantics segmentation, hereafter referred to as "point cloud segmentation". In the research of automating the creation of polyhedral building models, Rottensteiner [6] used the method of curvature segmentation to detect the point cloud in the entire area of the building, then extracted the roof plane and grouped the planes to complete the modeling. Pu S et al. [7] proposed aplanar surface growing algorithm to segment point clouds, set a threshold in the point set and select seed points, and then judge whether the points that meet the growth rules have grown to achieve segmentation of the same type of point cloud. However, the point cloud itself is a point set composed of a group of disordered points in Euclidean space, meanwhile, the amount of point cloud...
data is usually large, and there are often losses, noises, and incomplete in the data. In data processing, the traditional algorithm of manually labeling features is used for point cloud segmentation, and the workload will be very huge.

Qi et al. [8] proposed PointNet in 2017. This network is a pioneering work for deep neural networks to directly process point cloud data. It is a milestone achievement in 3D point cloud data processing. This paper proposes a deep-learning based method to create BIM from point cloud data, we first collects the Stanford large-scale 3D Indoor Spaces Dataset (S3DIS) data set [9], corrects the errors in the data set; then imports the data set into the PointNet network and conducts training, and continuously adjusts the hyperparameters and the structure of the network according to the results, seeks the optimal optimization effect, and obtains a reasonable point cloud data geometric parameter, and finally use the visual programming tool in Dynamo [10] to write a script that automatically creates BIM in Revit based on the geometric parameters, and input the parameters to create the BIM corresponding to the point cloud data.

2. Process of PointNet Semantic Segmentation

2.1. PointNet Network Framework

Figure 1 shows the network structure of PointNet classification and segmentation. First, there are n points \{P_i\}_{i=1,...,n} as input, expressed as an n*3 2D tensor, where n represents the number of point clouds, "3" represents the coordinates of the x, y, z axis corresponding to the point; the input data is multiplied with the transformation matrix learned by the T-net network for alignment, which ensures the invariance of the model to specific spatial transformations; then PointNet uses multiple mlp is performed to obtain local features of point cloud data, and the symmetric operation MaxPooling is used to learn global features in each dimension. At the same time, PointNet adds a normalization term to the softmax training loss to constrain the feature transformation matrix to be close to the orthogonal matrix:

$$L_{reg} = \| I - AA^T \|_F^2$$

A represents the feature alignment matrix predicted by the small network in the formula. By adding a regular term, the network optimization becomes more stable, and the model obtains better performance.

For classification tasks, PointNet uses global features to obtain the classification scores of k categories. The segmentation network is an extension of the classification network, which is highly shared with the structure of the classification network. When dealing with the segmentation problem,
the network combines the global feature with the 64-dimensional point-wise feature of the point cloud, so that the network can predict the number of points that depend on the local geometry and global semantics. Then output the probability of point-by-point classification through mlp. The network uses Batchnorm in all layers with ReLU, and the Dropout layer is used for the last mlp in the classification network.

In order to realize the semantic segmentation of point cloud data, this paper uses the S3DIS data set as experimental data, and tries and verifies the optimal model structure in the process of continuous training and parameter adjustment on the PointNet network, and finally selects the best effect on the verification set and the model structure as the final network result.

2.2. Point cloud Segmentation Results
In this paper, Area1, 2, 3, 4, 5 are used as training set and area 6 as test set in PointNet. Experimental records are shown in the figure 2 and 3 below:

![Figure 2. Train sets result](image1)

![Figure 3. Test sets result](image2)

The predict result of Area_6 Conference room_1 and Office_1 point cloud segmentation is as follows figures 4 and 5:
As can be seen from the above figure, the overall effect of PointNet segmentation is good, especially for the segmentation of walls, floors, ceilings and beams that compose the room structure, the segmentation accuracy of most points exceeds 95%. In terms of segmentation of doors, windows, tables and chairs, the performance of PointNet is still worthy of recognition, however, there are segmentation errors caused by insufficient classification on the boundary between components in contact with each other. For other smaller and more complex building components in the room, such as whiteboards and bookcases, the segmentation effect of PointNet is not satisfactory, and a large area of points is classified into the wrong label.

3. Modeling BIM with Dynamo

Dynamo is an open source plug-in based on Autodesk Revit, which provides a visual programming language (VPL) based on a stream interface [11]. To create a BIM, geometric parameters of the point cloud is been obtained firstly. Therefore, we compose all the point cloud vectors of the floor and ceiling into the vector group of this instance, and then obtain the coordinate values of all the point clouds on the x-axis, y-axis, and z-axis. The values are sorted to obtain the extreme values of the point cloud vector on the three coordinate axes, which are recorded as \{x_min,x_max\}, \{y_min,y_max\} and \{z_min,z_max\}. After obtaining the geometric information of the point cloud data, we import the obtained coordinates into Dynamo. Here we take the Area_6_Office_1 in the S3DIS data set as an example. The specific steps below are as shown in figure 6.
3.1. Floor and Ceiling

The floor elevation is set to 0 m, and then the x and y coordinates in the table are formed into 4 floor boundary points $f_1(-20.996, 33.314)$, $f_2(-15.384, 33.314)$, $f_3(-15.384, 36.237)$, $f_4(-20.996, 36.237)$, as shown in table 1, input the coordinates of these four boundary points through the code block node of Dynamo to read the coordinate information of the point; then use the Rectangle.ByCornerPoints node to generate a rectangular boundary line, and convert the boundary line to the PolyCurve node into aggregation Curve; finally, enter the Floor.ByOutlineTypeAndLevel node as the outline of the floor. At the same time, create the Floor Type node and the Levels node to enter the family type and the elevation of the floor. The family type should be imported into Revit in advance, otherwise Dynamo will report an error that the specified type cannot be found. Connecting the above three nodes to Floor.ByOutlineTypeAndLevel can generate floor BIM at the specified coordinate position. Since the S3DIS data set only obtains the point cloud data information of one side, the thickness of the floor, wall and other components cannot be accurately known. This paper sets the default thickness of the floor and wall to 200 mm.

Table 1. The three coordinate extreme values of the floor point cloud.

|       | x     | y     | z     |
|-------|-------|-------|-------|
| Min   | -20.996 | 33.314 | 0.000 |
| Max   | -15.384 | 36.237 | 0.001 |

The process of generating a ceiling BIM is roughly the same as generating a floor BIM. In this article, the walls are built in the vertical direction by default, so the x and y coordinates of their boundary points are the same. The main difference is that the z-axis coordinates of the ceiling and the floor are inconsistent. From the point cloud data, it can be seen that the elevation of the ceiling is located at 2.704 m. Similarly, the ceiling BIM can be generated.

3.2. Wall

The positioning line of the wall is created based on the boundary line of the floor. Use the boundary line of the floor in the previous section as the curve to generate the wall and enter the Wall.ByCurveAndHeight node, other inputs include the height of the wall, the Level of the bottom of the wall, and the Revit family type by FamilyType node, then run the nodes to create the wall BIM. It is worth noting that the wall height here is the same height as the ceiling, in fact, this is not the true wall height parameter value, because the existence of the ceiling causes part of the wall to be blocked.
by the ceiling, so the ceiling height cannot accurately reflect the actual wall height. For now, the occlusion problem is still a big challenge in the process of automatically creating an As-built BIM.

3.3. Beam and Column
The creation of beams and columns (as same as the creation of the beam) is simpler than the creation of floors and walls. Dynamo provides a node StructuralFramingBeamByCurve to create beam BIM based on the centerline and elevation of the beam. After the creation of the floor, ceiling and wall, all you need to do is to create a beam centerline at the corresponding position in Revit, then select the beam Level and beam family type by StructuralFramingType node to complete the beam BIM creation. The same as when creating a wall BIM, the beam size also has the problem of blocking and hiding the structure, so the beam section size \( b=519 \text{ mm}, h=562 \text{ mm} \) used in this article is derived from the positional relationship of the original point cloud data, which is also not completely correct reflect the actual section size of the beam.

3.4. Door, Window and Other Instance Families
There is no node that can directly generate doors&windows in Dynamo. Both doors&windows belong to family instance in Revit, but FamilyInstance nodes cannot be used directly to generate host-based family instances such as doors&windows in Dynamo, so it is necessary to use Revit API tools to achieve the work of creating doors&windows BIM. This article uses the Python Script node to create the door&window BIM by coding the Revit API in Dynamo. Other node inputs are the door&window family type, the specified door&window host, the center point coordinates of the door&window bottom and the door&window elevation. As for the rest of the instance families such as tables, chairs, whiteboard, bookcase and sofa, they can be created by FamilyInstance nodes directly, it just needs to provide the center point coordinates and the family type and the BIM will be created at once.

The final BIM model is obtained by combining the above steps as shown in figure 7 and figure 8 below:

![Figure 7. Office_1 BIM modeling](image7)

![Figure 8. Conference room_1 BIM modeling](image8)

4. Conclusion
From the results, whether the finally created BIM can accurately reflect the actual data in the point cloud data set mainly depends on the accuracy of the deep neural network in point cloud semantic segmentation, so improving the accuracy of network classification and segmentation is one of the most direct improvement methods of BIM modeling accuracy. At the same time, this problem is closer to the instance segmentation of point clouds rather than semantic segmentation, which is equivalent to divide the same label components into independent instance individuals on the basis of point cloud
semantic segmentation. Compared with semantic segmentation, instance segmentation is more challenging because it requires more precise and fine-grained point cloud reasoning [12], which will be our future the direction of the work.

Reference

[1] Xiong X, Adan A, Akinci B, et al. 2013 Automatic creation of semantically rich 3D building models from laser scanner data Automation in Construction 31 325-37.
[2] Barbosa M J, Pauwels P, Ferreira V, et al. 2016 Towards increased BIM usage for existing building interventions Structural Survey 34(2) 168-190.
[3] Chen W, Chen K, Cheng J C P, et al. 2018 BIM-based framework for automatic scheduling of facility maintenance work orders Automation in Construction 91 15-30.
[4] Díaz-Vilariño L, Lagüela S, Armesto J, et al. 2014 Indoor daylight simulation performed on automatically generated as-built 3D models Energy and Buildings 68 54-62.
[5] Huber D, Akinci B, Adan A, et al. 2011 Methods for automatically modeling and representing as-built building information models Proc. of NSF Engineering Research and Innovation Conference Atlanta Georgia.
[6] Rottensteiner F 2003 Automatic generation of high-quality building models from lidar data IEEE Computer Graphics and Applications 23(6) 42-50.
[7] Pu S and Vosselman G 2006 Automatic extraction of building features from terrestrial laser scanning ISPRS Commission V Symposium “Image Engineering and Vision Metrology” Dresden German 36 25-27.
[8] Charles R Q, Su H, Kaichun M, et al. 2017 PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Honolulu HI p 77-85.
[9] Armeni I, Sener O, Zamir AR, et al. 2016 3D semantic parsing of large-scale indoor spaces IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Las Vegas NV p 1534-1543.
[10] Autodesk, Inc. 2020 Dynamo Developer. http://www.dynamobim.org.
[11] Autodesk, Inc. 2018 http://paulaubin.com/downloads/2017AU/BIM128338/Aubin-AU2017.pdf
[12] Guo Y, Wang H, Hu Q, et al. 2020 Deep learning for 3D point clouds: a survey IEEE Transactions on Pattern Analysis and Machine Intelligence (99) p 1-1.