Point Surfel Transformer Network for Semantic Segmentation of Large-Scale ALS Point Clouds

XINLONG ZHANG[1], RUIHANG XUE[1], MICHAEL KÖLLE[1] & UWE SÖRGEL[1]

Abstract: Automated semantic segmentation of point clouds plays an important role in 3D scene perception. The definition of relevant features is usually key for segmentation and classification, with automated workflows presenting the main challenges. Point Transformer networks based on self-attention operator, which is invariant to permutation of the input elements, can describe the essential attributes of disordered point clouds well. However, the application to large-scale Airborne Laser Scanning (ALS) point cloud scenes is not trivial. Naive Point Transformer lacks the ability to describe local features, therefore most of the established methods focus on small simple scenes. In this work, the Point-Surfel Transformer (PS-Transformer) network based on not only point features but also local surfel features to strengthen the local perception, is proposed. Our approach is evaluated on the Hessigheim High-Resolution 3D Point Cloud (H3D) Benchmark and achieves state-of-the-art 89.19% overall accuracy. Furthermore, our proposed PS-Transformer approach outperforms outperforms naive point transformer by a large margin of 4.64 percentage points.

1 Introduction

Driven by the fast development of depth sensors and 3D scanners, the automatic processing of point clouds has played an indispensable role in the fields of mapping geographic information, autonomous driving and robotics (CHEN et al. 2020). As a key step of understanding 3D scenes, semantic segmentation of point clouds has attracted extensive attention from researchers.

Current methods can be generally grouped into two categories: the models based on handcrafted features and the models based on deep learning. In the first category, manually designed features require expensive calculation and suffer from poor generalization (LECUN et al. 2015). On the contrary, the data-driven features extraction based on deep learning does not need hand-designed feature extractors, and can automatically learn better representations of objects (LECUN et al. 2015). Compared with the regular grid structure of image data, the disordered distribution of point clouds makes 3D semantic segmentation a challenging task.

In recent years, a large number of models based on deep learning have been proposed for 3D semantic segmentation. Inspired by 2D convolutional neural networks (CNNs), VoxNet (MATURANA et al. 2015) voxelizes point clouds to make the data structure suitable for 3D CNNs. But the sparsity of point clouds leads to the low efficiency of voxel grid arrangement and the high computational burden. To alleviate this problem, the sparse convolutional network (LIU et al. 2015) and its 3D application (VERDOJA et al. 2017) operate only on voxels that are not empty. However, such methods only depend on the voxel boundary and ignore the local geometric structure. PointNet++ (QI

[1] University of Stuttgart, Institute for Photogrammetry,
Geschwister-Scholl-Str. 24D, D-70174 Stuttgart,
E-Mail: [Xinlong.Zhang, Ruihang.Xue, Michael.Kölle, Uwe.Sörgel]@ifp.uni-stuttgart.de

DOI: 10.24407/KXP:1796046701

et al. 2017) effectively solves the problem of extracting local features by combining sampling-grouping layer and PointNet (QI et al. 2017) layer. Due to the network's pooling operator, features in each individual dimension have the same weight. The self-attention based Point Transformer (ZHAO et al. 2021) weights each element adaptively. Due to this set operation the network is invariant to permutation of the input elements (JADERBERG et al. 2015), which is consistent with the disordered distribution of point clouds. Nevertheless, the relationship between points in sets is not taken into consideration.

In complex large-scale point clouds, naive Point Transformer usually fails to extract sufficiently effective semantic features and performs poorly, because of weak local features. Surface element (surfel) features on the other hand (PFISTER et al. 2000) comprise normal, curvature, scale and others, which can provide the attributes of the local approximate surface of each point (WEINMANN et al. 2013). This kind of geometric adjacency information can strengthen the network's perception of local regions.

In this work, the Point-Surfel Transformer (PS-Transformer) network, based on not only point features but also surfel features, is proposed. First, the local surfel features are generated from point coordinates. Then, the surfel features are mapped to the original point feature space to obtain the point-surfel features. Subsequently, the enhanced features are fed to the PS-Transformer network to generate point-wise feature vectors in an encoder-decoder architecture. Finally, a multi-layer perceptron (MLP) maps each feature vector to the final logits, which are used to predict the class probability of each point.

2 Point Surfel Transformer Network

2.1 Surfel Features Extraction

Surfel features of each individual point p are extracted from the raw XYZ coordinates of point clouds, which can be expressed by one 6-tuple

$$\mathrm{F}_p = (n_{px}, n_{py}, n_{pz}, d_p, f_p, r_p) \tag{1}$$

where n_{px}, n_{py}, n_{pz} are the parameters of the normal vector, d_p is the distance from the origin to the fitted plane of point p, f_p is change of curvature, and r_p is the residual from point p to its fitted surface. Normals distinguish flat and inclined surfaces, d_p is the association with global information, f_p represents the local surface variation and r_p describes the local roughness.

These local features of each point $p(x_p, y_p, z_p)$ are calculated as follows. Firstly, the covariance matrix $M_{3\times3}$ is constructed via the Eq. (2).

$$M_{3\times3} = \frac{1}{k}(p_i - \overline{p}) \cdot (p_i - \overline{p})^T \tag{2}$$

Where p_i is the K Nearest Neighbour(KNN) of point p, and $\overline{p}$ represents the mean vector of the point p and its KNNs. Then eigenvalues and eigenvectors of $M_{3\times3}$ are calculated by Singular Value Decomposition (SVD) as Eq. (3) and Eq. (4).

$$\lambda V = M_{3\times 3} V \tag{3}$$

$$\begin{cases} V = (v_1^{(M)}, v_2^{(M)}, v_3^{(M)}) \\ \lambda = diag(\lambda_1^{(M)}, \lambda_2^{(M)}, \lambda_3^{(M)}), \lambda_1^{(M)} > \lambda_2^{(M)} > \lambda_3^{(M)} \end{cases} \tag{4}$$

where λ is the vector of eigenvalues of the covariance matrix and V is the set of eigenvectors of the covariance matrix. The local features of point p are calculated as Eq. (5).

$$\begin{cases} n_p(n_{px}, n_{py}, n_{pz}) = v_3^{(M)} \\ d_p = -(n_{px}x_p + n_{py}y_p + n_{pz}z_p) \\ f_p = \dfrac{\lambda_3^{(M)}}{\lambda_1^{(M)} + \lambda_2^{(M)} + \lambda_3^{(M)}} \\ r_p = \dfrac{|n_{px}x_p + n_{py}y_p + n_{pz}z_p + d_p|}{\sqrt{n_{px}^2 + n_{py}^2 + n_{pz}^2}} \end{cases} \tag{5}$$

The extracted surfel features composed of low-level descriptors contain rich local information, which will be used for the following processes.

2.2 Point Transformer Networks

The overall structure of the proposed PS-Transformer is shown in Fig. 1, where the generated surfel features are firstly concatenated with the point features (coordinates and RGB values), and then a U-Net-like (RONNEBERGER et al. 2015) point Transformer is implemented for feature fusing and learning. The U-Net-like Transformer is composed of point transformer block, transition down and transition up modules. Connecting them alternately could obtain an 8-layer network, where the first 4 layers are encoders, and the last 4 layers are decoders. In the first layer, a multilayer perceptron (MLP) is applied to fuse the point-surfel features. Besides, each feature encoder layer has a point transformer block connected by a transition down module. Moreover, each feature decoder layer has a point transformer block connected by a transition up module. Finally, the encoder-decoder structure together forms a U-Net-like network to fulfill the semantic segmentation task.

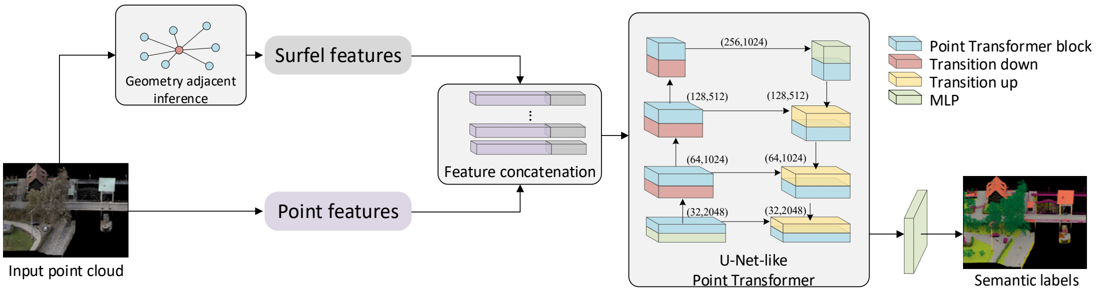


Fig. 1: Overall structure of the proposed PS-Transformer

As the core of the proposed network, the point transformer block is formed by cascading two linear mappings and a self-attention calculation. The linear mapping converts the input-output dimension, and the self-attention estimates the internal relationship among the input points. For the input points set X, the subset $X_i \in X$ is the neighborhood of the point x_i, which is obtained by the KNN algorithm, where $i = 1, 2, ..., N$ and the number of neighborhood points is N. Hence the self-attention calculation of the point x_i is defined as:

$$y_i = \sum_{x_j \in X_i} \alpha\left(\beta\left(\varphi_k\left(x_i\right) - \varphi_q\left(x_j\right) + \mathrm{pos}\right)\right) \mathrm{e} \;\; \varphi_v\left(x_j + \mathrm{pos}\right) \qquad (6)$$

where y_i is the output feature vector, α is the softmax activation function, and β is the attention mapping function, which is implemented by a MLP, i.e. 2 linear layers and a ReLU (GLOROT et al. 2011) activation function. φ_k, φ_q and φ_v are all linear mappings for adapting to different feature dimensions, and e denotes an elementwise multiplication. pos is the positional coding, which is a linear mapping from the relative coordinate of the points:

$$\mathrm{pos} = \varphi_p\left(cor_i - cor_j\right) \qquad (7)$$

where cor_i and cor_j are respectively the 3D coordinates of the point i, j, φ_p is a MLP.

Transition down module realizes the down-sampling of the local point cloud, which is implemented by the farthest point sampling and KNNs searching. Then the feature vectors of the sampled subset are obtained by local max-pooling. Transition up module realizes the up-sampling of the local point cloud, where the number of points is increased by trilinear interpolation. Then the features of the corresponding encoder layer are added to the up-sampled new point set, which is also the U-Net-like connection. The feature decoder has a symmetrical configuration with the encoder as shown in Fig. 1, where (32,2048) represents that in this layer, the feature dimension is 32 and the output point number is 2048.

For the semantic segmentation task, the PS-Transformer network should have an output label for each point. The 4-by-4 symmetrically designed encoder and decoder could exactly guarantee the correspondence between the input point-surfel features and the output labels. At the last layer, a MLP maps the point feature to the label space y_k, and all the learnable parameters of PS-Transformer could be updated by optimizing the cross-entropy loss function:

$$\mathrm{Loss}(\mathbf{w}) = -\sum_{i=1}^{N}\sum_{k=1}^{K} t_{ki} \ln y_k(X_i, \mathbf{w}) \qquad (8)$$

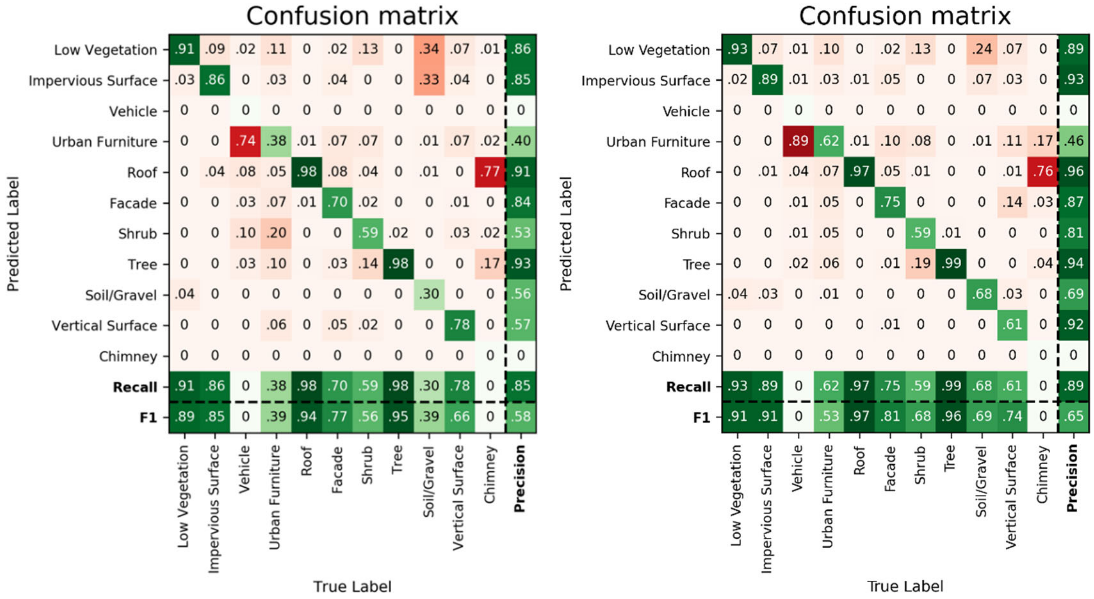
where K denotes to the number of categories, t_{ki} denotes to the one-hot truth corresponding to the i th point, and $\mathbf{w}$ denotes to the set of learnable parameters.

3 Experiments

In this section, we conduct comparative experiments to evaluate our PS-Transformer networks. The experiments are based on the Hessigheim High-Resolution 3D Point

Cloud (H3D) Benchmark (KÖLLE et al. 2021). The dataset was collected by a Riegl VUX-1LR Scanner and two oblique-looking Sony Alpha 6000 cameras integrated on a RIEGL Ricopter platform. The mean point density is 800 points/m² enriched by RGB colors and the ground sampling distance (GSD) of images is 2-3 cm. In addition, the points have been manually labeled with the following 11 classes: Low vegetation, Impervious surface, Vehicle, Urban furniture, Roof, Facade, Shrub, Tree, Soil/Gravel, Vertical surface, Chimney.

In order to reduce the computational burden, the training data and the test data are cropped into 49 splits and 22 splits, respectively. Our implementation of the PS-Transformer is realized in PyTorch. The Adam optimizer is employed in the network, and we train the network for 20 epochs with an initial learning rate of 0.0005. The segmentation results are evaluated by overall accuracy (OA), recall, and F1-score.



	Low Vegetation	Impervious Surface	Vehicle	Urban Furniture	Roof	Facade	Shrub	Tree	Soil/Gravel	Vertical Surface	Chimney	**Precision**
Low Vegetation	.91	.09	.02	.11	0	.02	.13	0	.34	.07	.01	.86
Impervious Surface	.03	.86	0	.03	0	.04	0	0	.33	.04	0	.85
Vehicle	0	0	0	0	0	0	0	0	0	0	0	0
Urban Furniture	0	0	.74	.38	.01	.07	.07	0	.01	.07	.02	.40
Roof	0	.04	.08	.05	.98	.08	.04	0	.01	0	.77	.91
Facade	0	0	.03	.07	.01	.70	.02	0	0	.01	0	.84
Shrub	0	0	.10	.20	0	0	.59	.02	0	.03	.02	.53
Tree	0	0	.03	.10	0	.03	.14	.98	0	0	.17	.93
Soil/Gravel	.04	0	0	0	0	0	0	0	.30	0	0	.56
Vertical Surface	0	0	0	.06	0	.05	.02	0	0	.78	0	.57
Chimney	0	0	0	0	0	0	0	0	0	0	0	0
Recall	.91	.86	0	.38	.98	.70	.59	.98	.30	.78	0	.85
F1	.89	.85	0	.39	.94	.77	.56	.95	.39	.66	0	.58

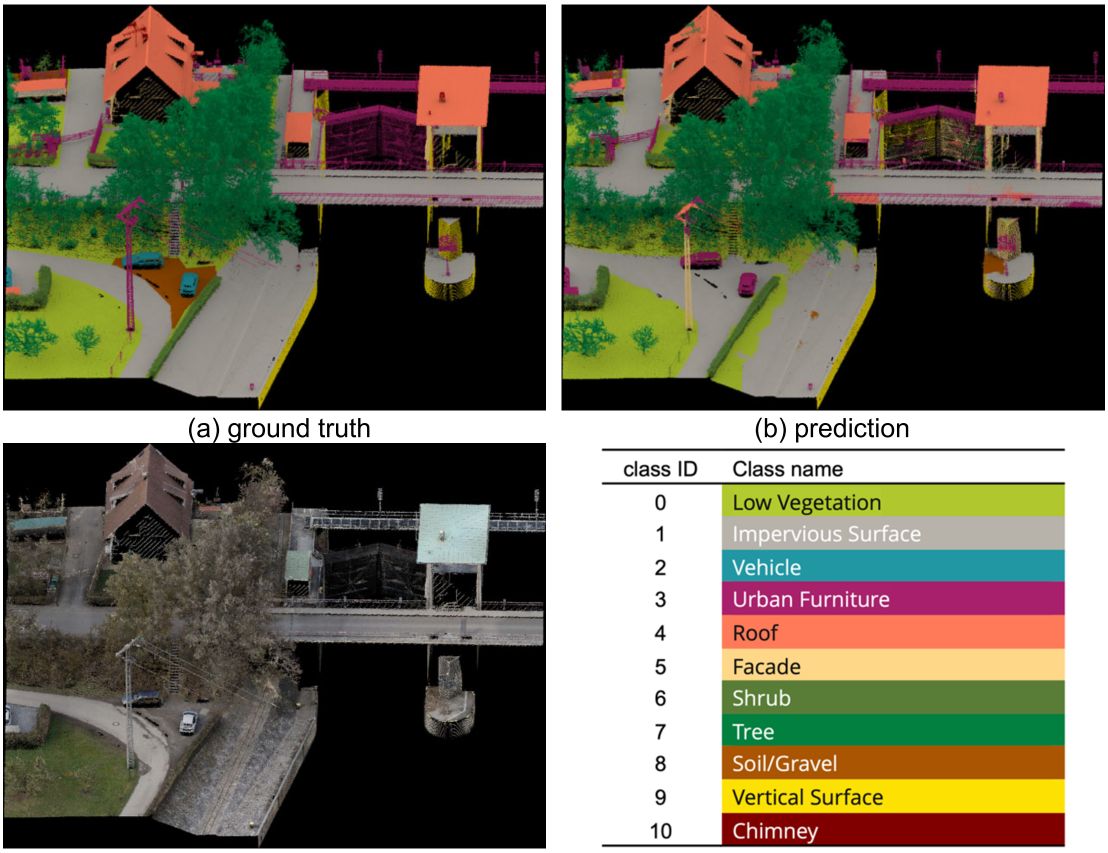
(a) naive Point Transfomer

	Low Vegetation	Impervious Surface	Vehicle	Urban Furniture	Roof	Facade	Shrub	Tree	Soil/Gravel	Vertical Surface	Chimney	**Precision**
Low Vegetation	.93	.07	.01	.10	0	.02	.13	0	.24	.07	0	.89
Impervious Surface	.02	.89	.01	.03	.01	.05	0	0	.07	.03	0	.93
Vehicle	0	0	0	0	0	0	0	0	0	0	0	0
Urban Furniture	0	0	.89	.62	.01	.10	.08	0	.01	.11	.17	.46
Roof	0	.01	.04	.07	.97	.05	.01	0	0	.01	.76	.96
Facade	0	0	.01	.05	0	.75	0	0	0	.14	.03	.87
Shrub	0	0	.01	.05	0	0	.59	.01	0	0	0	.81
Tree	0	0	.02	.06	0	.01	.19	.99	0	0	.04	.94
Soil/Gravel	.04	.03	0	.01	0	0	0	0	.68	.03	0	.69
Vertical Surface	0	0	0	0	0	.01	0	0	0	.61	0	.92
Chimney	0	0	0	0	0	0	0	0	0	0	0	0
Recall	.93	.89	0	.62	.97	.75	.59	.99	.68	.61	0	.89
F1	.91	.91	0	.53	.97	.81	.68	.96	.69	.74	0	.65

(b) our PS-Transfomer approch

Fig. 2: Detailed H3D test set segmentation results from the comparative experiment: (a) naive Point Transfomer (b) our PS-Transformer approach

We compare against the naive Point Transfomer to showcase the benefits of surfel features, and the results are shown in Fig. 2. Our approach achieves state-of-the-art 89.19% overall accuracy, which outperforms the naive Point Transformer (84.55%) by a large margin. Furthermore, the recall of fine-grained classes is greatly improved, the recall of urban furniture increases from 38.06% to 62.02%, and the recall of soil/gravel increases from 29.85% to 68.18%.



(a) ground truth

(b) prediction

(c) RGB texture

(d) class catalog

class ID	Class name
0	Low Vegetation
1	Impervious Surface
2	Vehicle
3	Urban Furniture
4	Roof
5	Facade
6	Shrub
7	Tree
8	Soil/Gravel
9	Vertical Surface
10	Chimney

Fig. 3: Comparison of our prediction and ground truth on H3D: (a) ground truth (b) prediction (c) RGB texture (d) class catalog

Fig. 3 displays an example of our prediction, ground truth and RGB texture. Although chimneys are interpreted as roofs more frequently, they are less often classified as facades or vertical surfaces. The difficulties mainly exist between vehicles and urban furniture, and soil/gravel are often inferred as impervious surface. The ambiguities are caused by their limited inter-class distances and scarce appearances.

4 Conclusion

We have presented PS-Transformer, a surfel features enhanced network for semantic segmentation of large-scale ALS point clouds. The proposed PS-Transformer utilizes geometric adjacency information to strengthen the network's local perception, and achieves state-of-the-art 89.19% overall accuracy on the H3D Benchmark. Future work will be focused on semantic segmentation for fine-grained objects (such as vehicles), while being aimed at models in the case of imbalanced samples.

5 Acknowledgements

The research work in this paper has been funded by China Scholarship Council (No. 202008080145).

6 References

CHEN, S., LIU, B., FENG, C., VALLESPI-GONZALEZ, C. & WELLINGTON, C., 2020: 3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception. IEEE Signal Processing Magazine, **38**(1), 68-86, https://doi.org/10.1109/MSP.2020.2984780.

GLOROT, X., BORDES, A. & BENGIO, Y., 2011: Deep sparse rectifier neural networks. In Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 315-323.

JADERBERG, M., SIMONYAN, K. & ZISSERMAN, A., 2015: Spatial transformer networks. Advances in neural information processing systems, 28, 2017-2025.

LECUN, Y., BENGIO, Y. & HINTON, G., 2015: Deep learning. Nature, **521**(7553), 436-444, https://doi.org/10.1038/nature14539.

LIU, B., WANG, M., FOROOSH, H., TAPPEN, M. & PENSKY, M., 2015: Sparse convolutional neural networks. IEEE conference on computer vision and pattern recognition, 806-814.

MATURANA, D. & SCHERER, S., 2015: Voxnet: A 3d convolutional neural network for real-time object recognition. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 922-928, https://doi.org/10.1109/IROS.2015.7353481.

KÖLLE, M., LAUPHEIMER, D., SCHMOHL, S., HAALA, N., ROTTENSTEINER, F., WEGNER, J. D. & LEDOUX, H., 2021: The Hessigheim 3D (H3D) Benchmark on Semantic Segmentation of High-Resolution 3D Point Clouds and Textured Meshes from UAV LiDAR and Multi-View-Stereo. ISPRS Open Journal of Photogrammetry and Remote Sensing, **1**, 11.

PFISTER, H., ZWICKER, M., VAN BAAR, J. & GROSS, M., 2000: Surfels: Surface elements as rendering primitives. 27th Annual Conference on Computer Graphics and Interactive Techniques, 335-342.

QI, C. R., SU, H., MO, K. & GUIBAS, L. J., 2017: Pointnet: Deep learning on point sets for 3d classification and segmentation. IEEE conference on computer vision and pattern recognition, 652-660.

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.

RONNEBERGER, O., FISCHER, P. & BROX, T., 2015: U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 234-241.

VERDOJA, F., THOMAS, D. & SUGIMOTO, A., 2017: Fast 3D point cloud segmentation using supervoxels with geometry and color for 3D scene understanding. IEEE International Conference on Multimedia and Expo (ICME), 1285-1290.

WEINMANN, M., JUTZI, B. & MALLET, C., 2013: Feature relevance assessment for the semantic interpretation of 3D point cloud data. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, **5**(W2), 1.

ZHAO, H., JIANG, L., JIA, J., TORR, P. H. & KOLTUN, V., 2021: Point transformer. IEEE/CVF International Conference on Computer Vision, 16259-16268.