PointSIFT: A SIFT-like Network Module for 3D Point Cloud Semantic Segmentation

Mingyang Jiang
Shanghai Jiao Tong University
jmydurant@sjtu.edu.cn

Yiran Wu
Shanghai Jiao Tong University
yiranwu@sjtu.edu.cn

Tianqi Zhao
Tsinghua University
zhaotq16@mails.tsinghua.edu.cn

Zelin Zhao
Shanghai Jiao Tong University
sjtuytc@sjtu.edu.cn

Cewu Lu
Shanghai Jiao Tong University
lucewu@sjtu.edu.cn

Abstract

Recently, 3D understanding research sheds light on extracting features from point cloud directly [22, 24], which requires effective shape pattern description of point clouds. Inspired by the outstanding 2D shape descriptor SIFT [15], we design a module called PointSIFT that encodes information of different orientations and is adaptive to scale of shape. Specifically, an orientation-encoding unit is designed to describe eight crucial orientations, and multi-scale representation is achieved by stacking several orientation-encoding units. PointSIFT module can be integrated into various PointNet-based architecture to improve the representation ability. Extensive experiments show our PointSIFT-based framework outperforms state-of-the-art method on standard benchmark datasets. The code and trained model will be published accompanied by this paper.

1. Introduction

3D point cloud understanding is a long-standing problem. Typical tasks include 3D object classification [33], 3D object detection [11, 21, 27] and 3D semantic segmentation [22, 24, 25]. Among these tasks, 3D semantic segmentation which assigns semantic labels to points is relatively challenging. Firstly, the sparseness of point cloud in 3D space makes most spatial operators inefficient. Moreover, the relationship between points is implicit and difficult to be represented due to the unordered and unstructured property of point cloud. In retrospect of previous work, several lines of solutions have been proposed to resolve the problem. In [19] handcrafted voxel feature is used to model geometric relationship, and in [20] 2D CNN features from RGBD images are extracted. Additionally, there is a dilemma between 2D convolution and 3D convolution: 2D convolution fails to capture 3D geometry information such as normal and shape while 3D convolution requires heavy computation.

Recently, PointNet architecture [22] directly operates on point cloud instead of 3D voxel grid or mesh. It not only accelerates computation but also notably improves the segmentation performance. In this paper, we also work on raw point clouds. We get inspiration from the successful feature detection algorithm Scale-invariant feature transform (SIFT) [15] which involves two key properties: scale-awareness and orientation-encoding. Believing that the two properties should also benefit 3D shape description, we design a novel module called PointSIFT for 3D understanding that possesses the properties. The main idea of PointSIFT is illustrated in Figure 1. Unlike SIFT algorithm which uses handcrafted features, our PointSIFT is a parametric deep learning module which can be optimized and adapted to point cloud segmentation.

The basic building block of our PointSIFT module is an orientation-encoding (OE) unit which convolves the features of nearest points in 8 orientations. In comparison to K-nearest neighbor search in PointNet++ [24] where K neighbors may fall in one orientation, our OE unit captures information of all orientations. We further stack several OE units in one PointSIFT module for representation of different scales. In order to make the whole architecture scale-aware, we connect these OE units by shortcuts and jointly optimize for adaptive scales. Our PointSIFT mod-

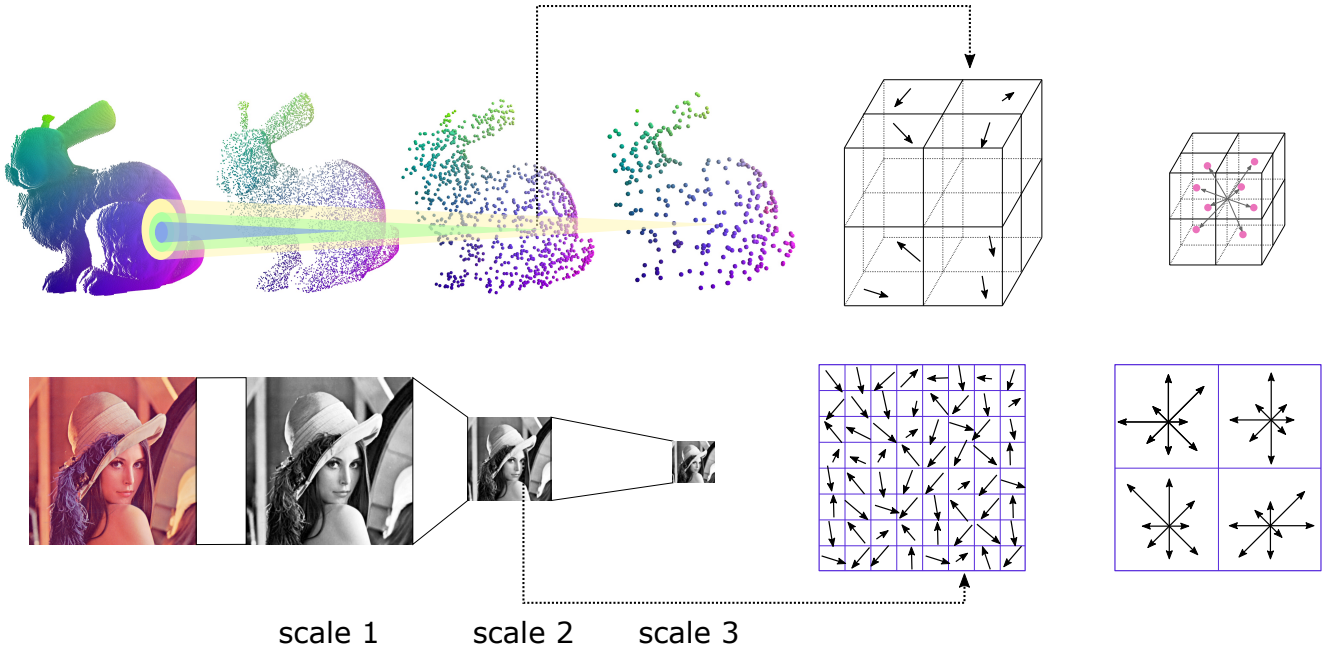

Figure 1. Structure of SIFT [15] and our PointSIFT module. The left side shows that both of them can capture multi-scale patterns and is adaptive to various scales. The right side shows that orientation is encoded in each key point/pixel.

ule receives point cloud with n features each point and outputs points of n features with better representation power. PointSIFT is a general module that can be integrated into various PointNet-based architectures to improve 3D shape representation.

We further build a hierarchical architecture that recursively applies the PointSIFT module as local feature descriptor. Resembling conventional segmentation framework in 2D [26] and 3D [24], our two-stage network first downsamples the point cloud for effective calculation and then upsamples to get dense predictions. The PointSIFT module is used in each layer of the whole framework and significantly improves the representation ability of the network.

Experimental results show that our architecture based on PointSIFT module outperforms state-of-the-art methods on S3DIS[1] (relative $\mathbf{12}\%$ improvement) and ScanNet[6] dataset(relative $\mathbf{8.4}\%$ mean IoU improvement).

2. Related Work

Deep learning on 3D data is a growing field of research. We investigate segmentation methods on several important 3D representations. Point cloud segmentation is discussed in more detail. After that, we briefly survey the SIFT descriptor where we borrow inspiration.

2.1. 3D Representation

Volumetric Representation The first attempt to apply deep learning is through volumetric representation. Many works [17, 23, 33] attempt to voxelize 3D point cloud or scene into regular voxel grids. However, the main challenges in volumetric representation are data sparsity and computational overhead of 3D convolution. A practical choice for resolution of voxel grid can be $32 \times 32 \times 32$, which is far from sufficient to faithfully represent complex shapes or scenes. Further, conversion is required to construct volumetric grids based on handy data format such as point cloud, which suffers from both truncation error and information loss. While some recent work [14, 25, 30] propose techniques to address the sparsity issue (*e.g.*, octree data structure), the performance of volumetric methods is still not on par with methods based on point cloud [23].

Polygonal Meshes Some literature [4, 16, 34] focuses on the use of graph Laplacian to process meshes. Further, functional map and cycle consistency [9, 10] helps build correspondence between shapes. However, this kind of methods are confined to manifold meshes.

Multi-view Representation [23, 29, 28] make an effort to exploit the strong capacity of 2D CNNs in 3D recognition. In these works, in order for the input to fit in 2D CNNs, the projection of 3D shapes to 2D images is required. The 3D-level understanding of object (or scene) is achieved by combining 2D images taken from various viewpoints. However, such kind of projection results in the loss of most crucial and discriminative geometric details. For example, calculating normal vectors becomes nontrivial, spatial distance is not preserved, and occlusion prevents a holistic under-

standing of both local and global structure. Failure to include those geometric details could substantially limit the performance in tasks such as shape completion and semantic segmentation.

2.2. Deep Learning on Point Cloud

PointNet and follow-up works Recently, a series of works propose several effective architectures that process point cloud directly. Among those, a big branch of works apply PoinetNet [22] as an unordered global or local descriptor. PointNet [22] is a pioneering effort that applies deep learning to unordered point clouds by point-wise encoding and aggregation through global max pooling. PointNet++ [24] proposes a hierarchical neural network to capture local geometric details. PointCNN [13] uses $\mathcal{X}$-Conv layer instead of vanilla mlp (multilayer perceptron) layer to exploit certain canonical ordering of points. Dynamic Graph CNN [32] (DGCNN) suggests an alternative grouping method to ball query used in PointNet++ [24]: KNN w.r.t. Euclidean distance between feature vectors. Superpoint Graphs[12] (SPG) first partitions point cloud into superpoints and embed every superpoint with shared PointNet. The semantic labels of superpoints are predicted from the PointNet embedding of current superpoint and spatially neighboring superpoints. While achieving leading results, we feel that segmentation algorithms could benefit from having ordered operators to some extent.

Rotation Equivariance and Invariance Another branch of work focuses on rotation equivariance or invariance. G-CNN [18] designs filter so that the filter set is closed under certain rotations (*e.g.*, 90-degree rotation) to achieve rotation invariance for those fixed rotations. This kind of method achieves exact rotation invariance only for some discrete rotations while introduces huge computational overhead. The time complexity is proportional to the cardinality of equivalence classes under the rotations one want to be equivariant of, making it impractical to consider rotation equivariance for large and general groups. Spherical CNN [5, 7] projects 3D shapes onto spheres and process the signal with spherical filters for rotation equivariant representation. Global max pooling that transforms sphere to a single value further achieves rotation invariance. While spherical CNNs is fully invariant to rotation, the projection of shapes onto sphere introduces large error and is not appropriate for objects with certain topological property or scenes. Besides, the operation that helps achieve rotation invariance substantially limits the model capacity, discouraging its use on segmentation tasks.

2.3. Scale-Invariant Feature Transform (SIFT)

SIFT [15] is a local image pattern descriptor widely used in object recognition, 3D modeling, robotics and various other fields. SIFT and its variants [3] consist of two entities, a scale-invariant detector and a rotation-invariant descriptor.

We borrow inspiration from both entities in SIFT algorithm. In keypoint detection stage, the SIFT algorithm achieves scale invariance with multi-scale representation which we also use for robustly processing objects of various scale. As for feature description stage, SIFT detects dominant orientations for rotation invariance and comprehensively perceives image pattern in different orientations. Given the fact that ordered descriptors like the descriptor of SIFT or kernels of CNN yield impressive results for 2D images, we expect that having such descriptors may also benefit representation of point cloud. The above two observations from SIFT algorithm lead us to design a scale-aware descriptor that encodes information from different orientations with ordered operations.

3. Problem Statement

We first formulate the task of point cloud semantic segmentation. The given point cloud is denoted as $\mathbf{P}$ which is a point set containing n points $p_1, p_2, ..., p_n \in \mathbb{R}^d$ with d dimensional feature. The feature vector of each point p_i can be its coordinate (x_i, y_i, z_i) in 3D space (or plus optional feature channels such as RGB values, normal, a representation vector in intermediate step, etc) . The set of semantic labels is denoted as $\mathbf{L}$. A semantic segmentation of a point cloud is a function Ψ which assigns semantic labels to each point in the point cloud. i.e.,

$$\Psi : \mathbf{P} \longmapsto L^n \tag{1}$$

The objective of segmentation algorithms are finding optimal function that gives accurate semantic labels.

Several properties of point set P have been emphasized in previous work [22, 24]. The density of P may not be uniform everywhere, and P can be very sparse. Moreover, P as a set is unordered and unstructured which distinguishes point cloud with common sequential or structured data like image or video.

4. Our Method

Our network follows a encode-decode (downsample-upsample) framework similar to general semantic segmentation network [2] for point cloud segmentation (Illustrated in Figure 2) . In the downsampling stage, we recursively apply our proposed PointSIFT module combined with set abstraction (SA) module introduced in [24] for hierarchical feature embedding. For upsampling stage dense feature is enabled by effectively interleaving feature propagation (FP) module [24] with PointSIFT module. One of our main contribution and core component of our segmentation network is the PointSIFT module which is endowed with the desired property of orientation-encoding and scale-awareness.

4.1. PointSIFT Module

Given an $n \times d$ matrix as input which describes a point set of size n with d dimension feature for every point, PointSIFT module outputs an $n \times d$ matrix that assigns a new d dimension feature to every point.

Inspired by the widely used SIFT descriptor, we seek to design our PointSIFT module as a local feature description method that models various orientations and is invariant to scale.

4.1.1 Orientation-encoding

Local descriptors in previous methods typically apply unordered operation (*e.g.*, max pooling [24, 32]) based on the observation that point cloud is unordered and unstructured. However, using ordered operator could be much more informative (max pooling discards all inputs except for the maximum) while still preserves the invariance to order of input points. One natural ordering for point cloud is the one induced by the ordering of the three coordinates. This observation leads us to the Orientation-encoding(OE) unit which is a point-wise local feature descriptor that encodes information of eight orientations.

The input of OE unit is a d-dimension feature vector $f_0 \in \mathbb{R}^d$ of point p_0. Information from eight orientations are integrated by a two-stage scheme to produce an orientation-aware feature f_0'. The OE Unit is illustrated in Figure 3.

The first stage of OE embedding is Stacked 8-neighborhood(S8N) Search which finds nearest neighbors in each of the eight octants partitioned by ordering of three coordinates. Since distant points provides little information for description of local patterns, when no point exists within searching radius r in some octant, we duplicate p_0 as the nearest neighbor of itself.

We further process features of those neighbors which resides in a $2 \times 2 \times 2$ cube for local pattern description centering at p_0. Many previous works ignore the structure of data and do max pooling on feature vectors along d dimensions to get new features. However, we believe that ordered operators such as convolution can better exploit the structure of data. Thus we propose orientation-encoding convolution which is a three-stage operator that convolves the $2 \times 2 \times 2$ cube along X, Y, and Z axis successively. Formally, the features of neighboring points is a vector V of shape $2 \times 2 \times 2 \times d$, where the first three dimensions correspond to three axes. Slices of vector M are feature vectors, for example $M_{1,1,1}$ represents the feature from top-front-right octant.The three-stage convolution is formulated as:

$$
\begin{aligned}
V_x &= g(Conv(W_x, V)) \in \mathbb{R}_{1\times2\times2\times d} \\
V_{xy} &= g(Conv(W_y, V_x)) \in \mathbb{R}_{1\times1\times2\times d} \\
V_{xyz} &= g(Conv(W_z, V_{xy})) \in \mathbb{R}_{1\times1\times1\times d}
\end{aligned}
$$

where $W_x \in \mathbb{R}_{2\times1\times1\times d}$, $W_y \in \mathbb{R}_{1\times2\times1\times d}$ and $W_z \in \mathbb{R}_{1\times1\times2\times d}$ are weights of convolution operator (bias is omitted for clarity). In this paper, we set $g(\cdot) = ReLU(\cdot)$. Finally, OE convolution outputs a d dimension feature by reshaping $V_{xyz} \in \mathbb{R}_{1\times1\times1\times d}$. Orientation-encoding Convolution (OEC) integrates information from eight spatial orientations and obtains a representation that encodes orientation information.

4.1.2 Scale-awareness

In order for our PointSIFT module to be scale-aware, we follow the long-standing method of multi-scale representation by stacking several Orientation-encoding (OE) units in PointSIFT module, as Figure 4 illustrated. Higher level OE units have larger receptive field than those of lower level. By constructing a hierarchy of OE units, we obtain a multi-scale representation of local regions in the point cloud. The features of various scales are then concatenated by several identity shortcuts and transformed by another point-wise convolution that outputs a d dimensional multi-scale feature. In the process of jointly optimizing feature extraction and the point-wise convolution that integrates multi-scale feature, neural networks will learn to select or attend to appropriate scales which makes our network scale-aware.

The fact that the input and output vector of our PointSIFT module is of the same shape makes it convenient for our module to be integrated into other existing point cloud segmentation architectures. We are looking forward to future applications of PointSIFT module, probably not restricted to point cloud segmentation domain.

4.2. Overall Architecture

We first revisit set abstraction (SA) and feature propagation (FP) module in PointNet++, thus present how to construct our model by SA, FP and pointSIFT modules.

4.2.1 Revisit SA and FP Module in PointNet++

Set abstraction (SA) and feature propagation (FP) modules are proposed in PointNet++ [24] that correspond to downsampling and upsampling of point cloud respectively. We give a very brief introduction of SA and FP module here.

A set abstraction module takes in $N \times d$ input standing for a point cloud of N points and d dimensional feature each point. The output is $N' \times d'$ which corresponds to N' downsampled points each with d' dimensional feature. The downsampling is implemented by finding N' centroids with farthest point sampling, assigning points to centroids and then calculate embedding of centroids by feeding feature of assigned points through shared PointNet.

The feature propagation module uses linear interpolation weighted by distances to upsample the point cloud. It receives input point set of size N and outputs an upsampled

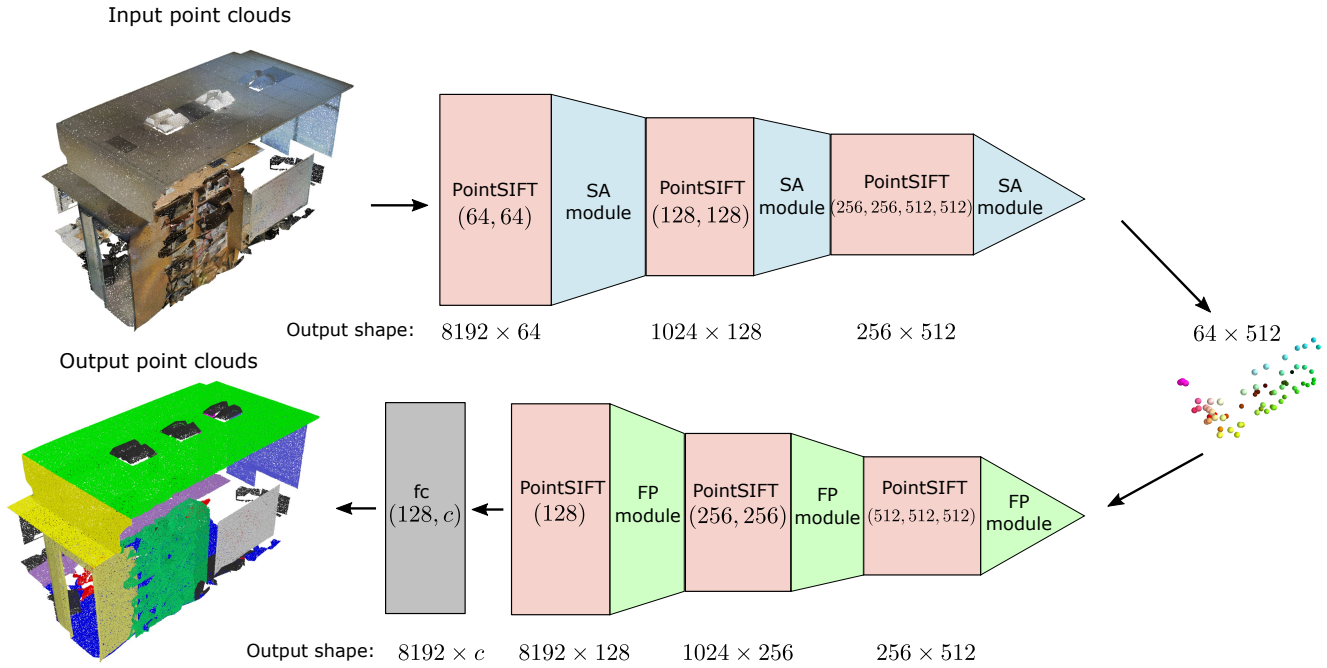

Figure 2. Illustration of our two-stage network architecture. The network consists of downsampling (set abstraction) and upsampling (feature propagation) procedures. PointSIFT modules (marked in red) are interleaved with downsampling (marked in blue) and upsampling (marked in green) layers. Both SA and FP module are introduced in [24]. The FP-shortcuts are not shown in the figure for better clarity. PointSIFT$(\cdot)$ specifies feature dimensionalities of each orientation-encoding(OE) units, for example, PointSIFT$(64, 64)$ stands for two stacked OE units both having 64 output feature channels. The number beneath layers is the shape of output point set of corresponding layers, for example, 8192×64 means 8192 points with 64 feature channel each point.

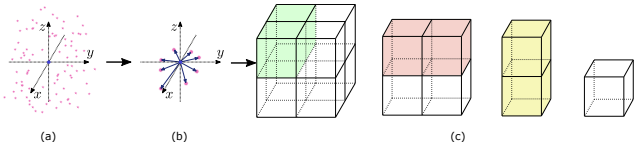

Figure 3. Illustration of Orientation-encoding(OE) Unit. (a): Point cloud in 3D space, the input point is at origin. (b) nearest neighbor search in eight octants. (c) convolution along X, Y, Z axis.

set of size N' where feature dimensionality is kept same. The upsampling process takes points that are dropped during downsampling, and assign features to them based on features of k nearest points that are not dropped weighted by Euclidean distance in 3D space.

4.2.2 Architecture Details

The input of our architecture is the 3D coordinates (or concatenating with RGB value) of 8192 points. In the downsampling stage, following [24], multi-layer perceptron (MLP) is used to transform the input 3D (or 6D) vectors into features with 64 dimensions. Three consecutive downsampling (set abstraction, SA) operations shrink the size of the point set to 1024, 256, 64 respectively. For the upsampling part, we use feature propagation(FP) module as proposed

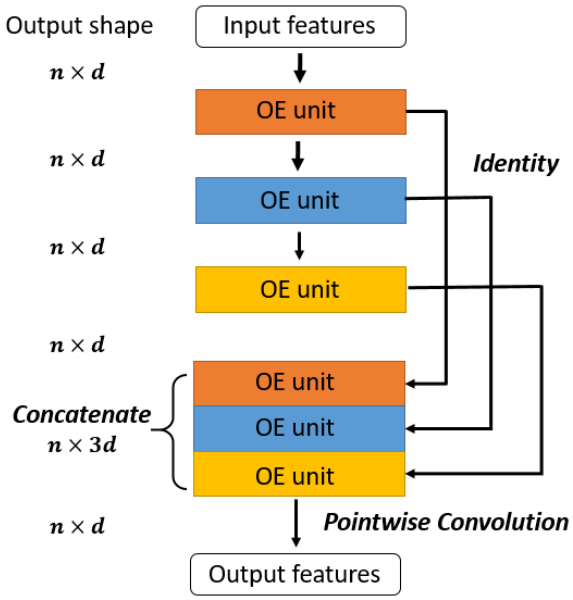

Figure 4. PointSIFT Module. Input features first pass through a series of Orientation-encoding(OE) layers, then outputs of OE units are concatenated and transformed by another point-wise convolution to obtain multi-scale feature.

Table 1. Effectiveness of PointSIFT Module.

downsampling step	first	second	third	fourth
point cloud size	8192	1024	256	64
captured point cloud size of Pointnet++[24]	6570	1010	255	64
captured point cloud size of PointSIFT framework	8192	1024	256	64

by [24] for dense feature and prediction. The point set is lifted to 256, 1024, 8192 points respectively by three FP layers which are aligned to its counterpart in the downsampling stage. Our PointSIFT module is inserted between all adjacent SA and FP layers. Finally, point features of the last upsampling layer pass through a fully connected layer for semantic label prediction.

Moreover, we insert FP modules not only between downsampling layers but also from downsampling layers to its counterpart in the encoding stage. We call these links FP-shortcuts for they resemble shortcuts by linking corresponding downsampling and upsampling layers and complements low-level information that might be lost during downsampling. The FP-shortcuts lead to much faster convergence which is proved by many prior works [13, 24] that use such shortcut flavor connections . The prediction accuracy is also improved by a considerable margin which is also reported in many works, *e.g.*, residual networks [8].

5. Experiments

Our experiment consists of two parts: verifying the effectiveness of the OE unit and PointSIFT module (Section 5.1) and introducing the results on semantic segmentation benchmark datasets (Section 5.2). In what follows, SA and FP are set abstraction and feature propagation module respectively, which are proposed by [24].

5.1. Effectiveness of PointSIFT Module

Orientation-encoding Convolution (OEC) We apply stacked 8-neighborhood search in OE unit, which is fundamentally different from ball query search proposed in PointNet++ [24]. The main difference lies in the fact that S8N search finds neighbors in each of 8 octants, while ball query searches for global nearest neighbors. As Figure 7 suggests, the global nearest neighbor search can result in a neighbor set of homogeneous points which is less informative than searching in 8 directions.

To justify our S8N search, we substitute ball query with S8N search in a lightweight version of PointNet++ [24] and compare the performance. The detailed architecture of the network is elaborated in Table 3. For fairness, we set the number of neighbors found by the two neighbor search method to be the same. The results are shown in Figure 6 which demonstrates the effectiveness of our S8N grouping plus OE Convolution.

Another observation that justifies the PointSIFT module is that more points from individual input point cloud contribute to the final representation for PointSIFT. In PointNet++ [24], the grouping layer fails to assign some points in the point cloud to any centroid and thus lost the information from the unassigned points. We claim the PointSIFT Module can almost avoid information loss in the downsampling process, which is beneficial to semantic segmentation. To prove this, we conduct an experiment on ScanNet [6] dataset compared to PointNet++ [24]. Given an input of size 8192, the first step is downsampling 8192 points to 1024 points. The method of [24] selects 1024 centroids and groups 32 nearest points inside the searching radius.

Our results show that in PointNet++[24], 1622 points on average are not grouped in 8192 points. That is, information of about 20% points are not integrated into the downsampled representation. On the contrary, our PointSIFT module involves more points in computation by performing multiple orientation-encoding(OE) convolutions in OE units. By inserting PointSIFT module before each downsampling layer, our results show that all points can be processed and make a contribution to final predictions. The results are reported in Table 1.

5.1.1 Effectiveness of Scale Awareness

We design a toy experiment to verify the effectiveness of scale-awareness in our framework. The experiment setting is that we generate 10000 simple shapes (*e.g.*, spheres, cuboids) with different scales, train our framework on generated data. Then we test if the activation magnitude of PointSIFT modules in different layer for certain shape is aligned to the scale of the shape. As aforementioned, different layers in PointSIFT module correspond to different scales. So if such alignment exists we can conclude that the network is aware of scale. It turns out that 89% of the time, the position of PointSIFT module with the highest activation in the hierarchy is aligned with the relative scale of input shape w.r.t max and min scale. This toy experiment demonstrates that the proposed PointSIFT framework is aware of scale in some sense.

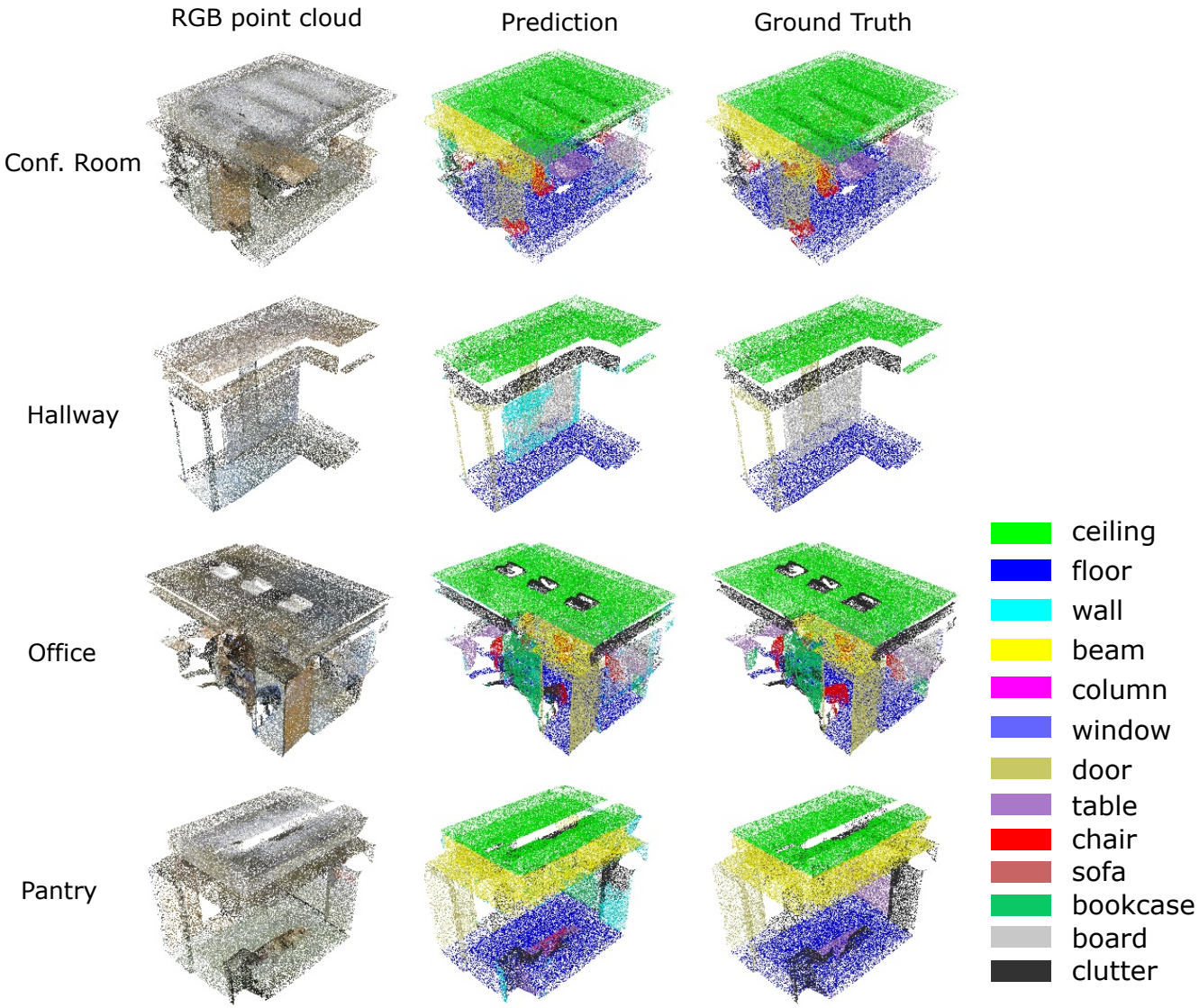

Figure 5. Visualization of results on S3DIS dataset[1]

Table 2. IoU for all categories of S3DIS[1] dataset.

Method	ceiling	floor	wall	beam	column	window	door	chair	table	bookcase	sofa	board	clutter
PointNet[22]	88.0	88.7	69.3	42.4	23.1	47.5	51.6	42.0	54.1	38.2	9.6	29.4	35.2
SegCloud[31]	90.06	96.05	69.86	0.00	18.37	38.35	23.12	**75.89**	70.40	58.42	40.88	12.96	41.60
SPGraph[12]	89.9	95.1	76.4	**62.8**	**47.1**	55.3	68.4	73.5	69.2	**63.2**	45.9	8.7	52.9
Ours	**93.7**	**97.9**	**87.5**	59.3	31.0	**73.7**	**80.7**	75.1	**78.7**	40.8	**66.3**	**72.2**	**65.1**

5.2. Results on Semantic Segmentation Benchmark Datasets

ScanNet ScanNet [6] is a scene semantic labeling task with a total of 1513 scanned scenes. We follow [24, 13], use 1201 scenes for training and reserve 312 for testing without RGB information. The result is reported in Table 4. Compared with other methods, our proposed PointSIFT method achieves better performance in the sense of per-voxel accuracy. Moreover, our method outperforms PointNet++ even though we do not use multi-scale grouping (MSG) in SA module which helps address the issue of non-uniform sampling density. The result demonstrates the advantage of our PointSIFT module in point searching and grouping.

layer name	output size	baseline model	ball query sampling	PointSIFT sampling
conv_1	1024×128	SA module		
conv_2	256×256	SA module	ball query sampling point-wise convolution SA module	PointSIFT module (128, 128) SA module
conv_3	64×512	SA module	ball query sampling point-wise convolution SA module	PointSIFT module (256, 256) SA module
pf_conv_3	256×512	FP module	FP module ball query sampling point-wise convolution	FP module PointSIFT module (512, 512)
pf_conv_2	1024×256	FP module	FP module ball query sampling point-wise convolution	FP module PointSIFT module (256, 256)
pf_conv_1	8192×128	FP module		
fc	8192×21	fully connected layer		

Table 3. Architectures for comparison of different sampling methods. After ball query sampling, point-wise convolution takes 32×1 kernels for extracting features.

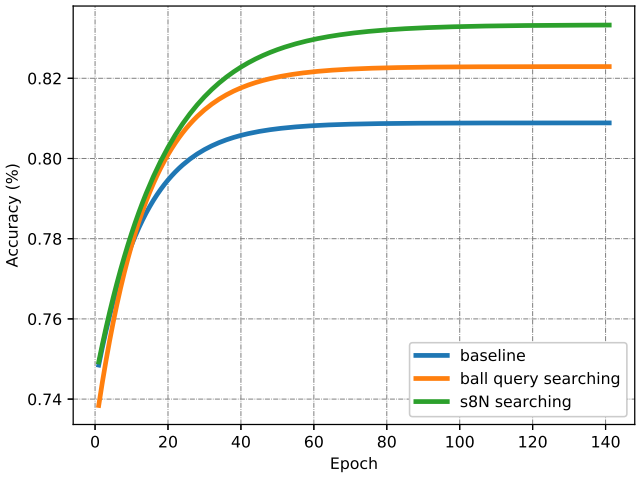

Figure 6. Accuracy for different searching methods. We use SavitzkyGolay filter for smoothing all the lines.

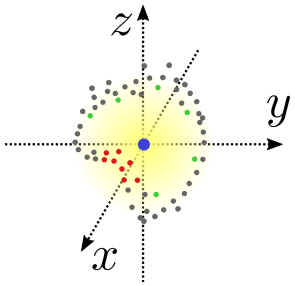

Figure 7. In this case, using K nearest neighbors, all chosen points are from one direction (red points). If we select points in different directions (green points), the representation ability will be better.

Stanford Large-Scale 3D Indoor Spaces The S3DIS dataset[1] takes six folders of RGB-D point cloud data from three different buildings (including 271 rooms). Each point is annotated with labels from 13 categories. We prepare the training dataset following [22] : we split points by room and sample rooms into 1m $\times$ 1m blocks. As have been used in [1, 22], we use k-fold strategy for train and test. Overall Accuracy and mIoU are shown in Table 5. Our PointSIFT architecture outperforms other methods. Per-category IoUs are shown in Table 2, our method improves remarkably on results of semantic segmentation task for this dataset and wins in most of the categories. Our method can achieve great results in some hard categories that other methods perform poorly, *i.e.* about 11 mIoU points on the sofa and 42 mIoU points on board. Some results are visualized in Figure 5.

Table 4. ScanNet[6] label accuracy and mIoU

Method	Accuracy %	mean IoU
3DCNN[6]	73.0	-
PointNet[22]	73.9	-
PointNet++[24]	84.5	38.28
PointCNN[13]	85.1	-
Ours	**86.2**	**41.5**

Table 5. Overall accuracy and meaning intersection over union metric of S3DIS[1] dataset.

Method	Overall Accuracy (%)	mean IoU (%)
PointNet[22]	78.62	47.71
SegCloud[31]	-	48.92
SPGraph[12]	85.5	62.1
PointCNN[13]	-	62.74
Ours	**88.72**	**70.23**

6. Conclusion

We propose a novel PointSIFT module and demonstrated significant improvement over state-of-art for semantic segmentation task on standard datasets. The proposed module is endowed with two key properties: First, orientation-encoding units capture information of different orientations. Second, multi-scale representation of PointSIFT modules enables the processing of objects with various scale. Moreover, an effective end-to-end architecture for point cloud semantic segmentation is proposed based on PointSIFT modules. We also conduct comprehensive experiments to justify the effectiveness of the proposed PointSIFT modules.

References

[1] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2016. 2, 7, 8

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006. 3

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *CoRR*, abs/1312.6203, 2013. 2

[5] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *CoRR*, abs/1801.10130, 2018. 3

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 6, 7, 8

[7] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. 3d object classification and retrieval with spherical cnns. *CoRR*, abs/1711.06721, 2017. 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 6

[9] Qi-Xing Huang and Leonidas J. Guibas. Consistent shape maps via semidefinite programming. *Comput. Graph. Forum*, 32(5):177–186, 2013. 2

[10] Qixing Huang, Fan Wang, and Leonidas J. Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Trans. Graph.*, 33(4):36:1–36:11, 2014. 2

[11] J. Lahoud and B. Ghanem. 2d-driven 3d object detection in rgb-d images. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4632–4640, Oct 2017. 1

[12] Loïc Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. *CoRR*, abs/1711.09869, 2017. 3, 7, 8

[13] Y. Li, R. Bu, M. Sun, and B. Chen. PointCNN. *ArXiv e-prints*, January 2018. 3, 6, 7, 8

[14] Yangyan Li, Soeren Pirk, Hao Su, Charles R Qi, and Leonidas J Guibas. Fpnn: Field probing neural networks for 3d data. *arXiv preprint arXiv:1605.06240*, 2016. 2

[15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004. 1, 2, 3

[16] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 832–840, Dec 2015. 2

[17] D. Maturana and S. Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *IROS*, 2015. 2

[18] Mahyar Najibi, Mohammad Rastegari, and Larry S. Davis. G-CNN: an iterative grid based object detector. *CoRR*, abs/1512.07729, 2015. 3

[19] J. Papon, A. Abramov, M. Schoeler, and F. Wrgtter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2027–2034, June 2013. 1

[20] Trung Pham, Thanh-Toan Do, Niko Sünderhauf, and Ian D. Reid. Scenecut: Joint geometric and object segmentation for indoor scenes. *CoRR*, abs/1709.07158, 2017. 1

[21] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017. 1

[22] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016. 1, 3, 7, 8

[23] Charles R Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. *arXiv preprint arXiv:1604.03265*, 2016. 2

[24] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 1, 2, 3, 4, 5, 6, 7, 8

[25] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[26] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *PAMI*, 2016. 2

[27] Shuran Song and Jianxiong Xiao. Deep Sliding Shapes for amodal 3D object detection in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[28] H. Su, F. Wang, E. Yi, and L. Guibas. 3d-assisted feature synthesis for novel views of an object. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2677–2685, Dec 2015. 2

[29] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[30] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[31] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. *CoRR*, abs/1710.07563, 2017. 7, 8

[32] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *CoRR*, abs/1801.07829, 2018. 3, 4

[33] Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, June 2015. 1, 2

[34] Li Yi, Hao Su, Xingwen Guo, and Leonidas Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. *arXiv preprint arXiv:1612.00606*, 2016. 2