Unsupervised Segmentation for Terracotta Warrior with Seed-Region-Growing CNN (SRG-Net)

Yao Hu †
School of Information Science and Technology, Northwest University, Xi'an 710127, Shaanxi, China
yhu6589@outlook.com

Wei Zhou †
School of Information Science and Technology, Northwest University, Xi'an 710127, Shaanxi, China
mczhouwei12@gmail.com

Guohua Geng
School of Information Science and Technology, Northwest University, Xi'an 710127, Shaanxi, China, gh
geng@nwu.edu.cn

Kang Li*
School of Information Science and Technology, Northwest University, Xi'an 710127, Shaanxi, China
likang@nwu.edu.cn

Xingxing Hao
School of Information Science and Technology, Northwest University, Xi'an 710127, Shaanxi, China
ystar1991@126.com

Xin Cao
School of Information Science and Technology, Northwest University, Xi'an 710127, Shaanxi, China
xin_cao@163.com

ABSTRACT

The repairing work of terracotta warriors in Emperor Qinshihuang Mausoleum Site Museum is handcrafted by experts, and the increasing amounts of unearthed pieces of terracotta warriors make the archaeologists too challenging to conduct the restoration of terracotta warriors efficiently. We hope to segment the 3D point cloud data of the terracotta warriors automatically and store the fragment data in the database to assist the archaeologists in matching the actual fragments with the ones in the database, which could result in higher repairing efficiency of terracotta warriors. Moreover, the existing 3D neural network research is mainly focusing on supervised classification, clustering, unsupervised representation, and reconstruction. There are few pieces of researches concentrating on unsupervised point cloud part segmentation. In this paper, we present SRG-Net for 3D point clouds of terracotta warriors to address these problems. Firstly, we adopt a customized seed-region-growing algorithm to segment the point cloud coarsely. Then we present a supervised segmentation and unsupervised reconstruction networks to learn the characteristics of 3D point clouds. Finally, we combine the SRG algorithm with our improved CNN (convolution neural network) using a refinement method. This pipeline is called SRG-Net, which aims at conducting segmentation tasks on the terracotta warriors. Our proposed SRG-Net is evaluated on the terracotta warrior data and ShapeNet dataset by measuring the accuracy and the latency. The experimental results show that our SRG-Net outperforms the state-of-the-art methods. Our code is available at https://github.com/hyoau/SRG-Net.

*Kang Li: Corresponding author
† Yao Hu and Wei Zhou contribute equally to this work

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487092

CCS CONCEPTS

• **Computing methodologies**; • **Machine learning**;

KEYWORDS

Point cloud, Unsupervised segmentation, Terracotta warrior, Convolution neural network, Seed region growing

ACM Reference Format:
Yao Hu †, Wei Zhou †, Guohua Geng, Kang Li, Xingxing Hao, and Xin Cao. 2021. Unsupervised Segmentation for Terracotta Warrior with Seed-Region-Growing CNN (SRG-Net). In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3487075.3487092

1 INTRODUCTION

Nowadays, the repairing work of terracotta warriors is mainly accomplished by the handwork of archaeologists in Emperor Qinshihuang's Mausoleum Site Museum. While more and more fragments of terracotta warriors are excavated from the site, the restoration does not have a high repairing efficiency. Moreover, the lack of archaeological technicians repairing the terracotta warriors makes it more uneasy to catch up with excavation speed. As a result, more and more fragments excavated from the archaeological site remain to be repaired. Terracotta Warrior is a 3D natural structure and can be represented with a point cloud. There are few pieces of researches concentrating on unsupervised point cloud part segmentation. Our work aims at simplifying the work of researchers in the museum, making it easy to match fragments excavated from the site with the pre-segmented fragments in the database. To improve the efficiency of the repairing work, we propose an unsupervised 3D point cloud segmentation method for the terracotta warrior data based on convolution neural network.

Our terracotta warrior data (see samples in Figure 1) is represented with $\{x_n \in R^p\}_{n=1}^N$, where R^p is the feature space, x_n means the features of one point, such as x, y, z, N_x, N_y, N_z(xyz coordinates and normal value), N means the number of points in one terracotta warrior 3D object. Our goal is to design a function $f : R^p \rightarrow L$, where L means the segmentation mapping labels, where $\{c_n \in L\}_{n=1}^N$ and c_n is each point label after the segmentation. In our problem, $\{c_n\}$ is

Figure 1: Simplified Terracotta Warrior Point Clouds.

fixed while the mapping function f and $\{x_n\}$ are trainable. To solve the unsupervised segmentation problem, we can split the problem into two sub-problems. Firstly, we need to design an algorithm to predict optimal L. Secondly, we need to design a network to learn the labels $\{c_n\}$ better and make full use of features of terracotta warrior point cloud. To get the optimal cluster labels $\{c_n\}$, we think that a good point cloud segmentation acts like what a human does. Intuitively, the points with similar features or semantic meanings are easier to segment to the same cluster. In 2D images, spatially continuous pixels tend to have similar colors or textures. In 3D point cloud, grouping spatially continuous points with similar normal value or colors are reasonable to be given the same label in 3D point cloud segmentation. Besides, the Euclidean distance between the points with the same label will not be very long. In conclusion, we design two criteria to predict the $\{c_n\}$:

·Points with similar spatial features are desired to be given the same label.

·The Euclidean distance between spatially continuous points should not be quite long.

An SRG algorithm is designed by us for learning the normal feature of point clouds as much as possible. With the segmentation method proposed in this paper, the complete terracotta warriors can be segmented into fragments corresponding to various parts automatically, e.g., head, hand, and foot, etc. The experiment [1] shows that the encoder-decoder structure of FoldingNet [2] does help neural network learn from point cloud. And DG-CNN is good at learning local features. So we propose our SRG-Net inspired by FoldingNet [2] and DGCNN [3] to learn part features of terracotta warriors better.

In Section 3, we describe the details of the process of assigning labels $\{c_n\}$. Section 4 will show that our method achieves better methods compared with other methods. Compared with existing methods, the key contribution of our work can be summarized as follows:

- We design a novel SRG algorithm for point cloud segmentation to make the most of point cloud xyz coordinates by estimating normal features.
- We propose our CNN inspired by DG-CNN and FoldingNet to learn the local and global features of 3D point clouds better.
- We combine the SRG algorithm and CNN neural network with our refinement method and achieve state-of-the-art segmentation results on terracotta warrior data.
- Our end-to-end model not only can be used in terracotta warrior point clouds but also can achieve quite good results on other point clouds. We also evaluate our SRG-Net on ShapeNet dataset.

2 RELATED WORK

Segmentation is typical both in the 2D image and 3D point cloud processing. In image processing, segmentation accomplishes a task that assigns labels to all the pixels in one image and clusters them with their features. Similarly, point cloud segmentation is assigning labels to all the points in a point cloud. The expected result is the points with the similar feature are given with the same label.

In the field of 3D point cloud segmentation, 3D point cloud segmentation problems can be classified into semantic segmentation, instance segmentation, and object segmentation.

As to semantic segmentation, semantic segmentation aims at separating a point cloud into several parts with the semantic meaning of each point. There are four main paradigms in semantic segmentation, including projection-based methods, discretization-based methods, point-based methods, and hybrid methods. Projection-based methods always project a 3D point cloud to 2D images, such as multi-view [4, 5], spherical [6, 7]. Discretization-based methods usually project a point cloud into a discrete representation, such as volumetric [8] and sparse permutohedral lattices [9, 10]. Instead of learning a single feature on 3D scans, several methods are trying to learn different parts from 3D scans, such as [9–11].

The point-based network can directly learn features on a point cloud and separate them into several parts. Point clouds are irregular, unordered, and unstructured. PointNet[12] is a pioneer that directly learns on the point cloud. A series of point-based networks has been proposed based on PointNet. However, PointNet can only learn features on each point instead of on the local structure. So PointNet++ is presented to get local structure from the neighborhood with a hierarchy network [13]. PointSIFT [14] is proposed to encode orientation and reach scale awareness. Instead of using K-means to cluster and KNN (K Nearest Neighbor) to generate neighborhoods like the grouping method PointNet++, PointWeb [15] is proposed to get the relations between all the points constructed in a local fully-connected web. As to convolution-based method, PointConv [16] uses the existing algorithm, using a Monte Carlo estimation to define the convolution. PointCNN [10] uses $\chi - conv$ transformation to convert the point cloud into a latent and canonical order. As to point convolution methods, Parametric Continuous Convolutional Neural Network (PCCN) [17] is proposed based on parametric continuous convolution layers, whose kernel function is parameterized by MLPs and spans the continuous vector space. Graph-based methods can better learn the features like shapes and geometric structures in point clouds. Graph Attention Convolution (GAC) [18] can learn several relevant features from local neighborhoods by dynamically assigning attention weights to

points in different neighborhoods and feature channels. Dynamic Graph CNN(DG-CNN) [3] constructs several dynamic graphs in neighborhood, and concatenates the local and global features to extract better features and update them each graph after each layer of the network dynamically. FoldingNet adopts the auto-encoder structure to encode the point cloud $N \times 3$ to 1×512 and decode it to $M \times 3$ with the aid of chamfer loss to construct the auto-encoder network.

3 PROPOSED METHOD

In this paper, our input data for the terracotta warrior is in the form of 3D point clouds (see samples in Figure 1). Point cloud data is represented as a set of 3D points $\{P_i \mid i = 1,\ 2,\ 3 \ldots n\}$, where each point is a vector R^n containing coordinates $x,\ y,\ z$ and normal features. Our method contains three steps:

1. We compute normal value with the xyz coordinates.
2. We use our seed-region-growing (SRG) method to pre-segment the point cloud.
3. We use our pointwise CNN called SRG-Net self-trained to segment the point cloud with the refinement of pre-segmentation results.

Given the point cloud only with 3D coordinates, there are several effective normal estimation methods like [19] [20] [21]. In our method, we follow the simplest method in [22] because of its low average-case complexity and quite high accuracy. We propose an unsupervised segmentation method for the terracotta warrior point cloud by combining our seed-region-growing clustering method with our 3D pointwise CNN called SRG-Net. The problem we want to solve can be described as follows. Firstly, the SRG algorithm is implemented on the point cloud to do pre-segmentation. Secondly, we use SRG-Net for unsupervised segmentation with pre-segmentation labels.

3.1 Seed Region Growing

Unlike 2D images, not all point cloud data has features such as color and normal. For example, our terracotta warrior 3D objects do not have any color feature because of the confidentiality clause. However, normal vectors can be calculated and predicted by point coordinates [23]. It is worth noting that there are many similarities and differences between color in 2D and normal features in 3D. For color features in a 2D image, if the pixels are semantically related, the color of the pixel in the neighborhood generally does not change a lot. For 3D point cloud normal features, compared with the color features in 2D images, the normal values of points in the neighborhood of the point cloud often differ. Points of neighborhood in one point cloud share similar features.

A seed-region-growing method is designed by us to pre-segment the point cloud based on the above characteristics of the normal feature of the point cloud. First, we implement KNN to the point cloud to get the nearest neighbors of each point. Then we initialize a random point as the start seed and add to the available points to start the algorithm. Then we choose the first seed from the available list to judge the points in its neighborhood. If the normal value and Euclidean distance are within the threshold we set, we think that the two points are semantically continuous, and we can group

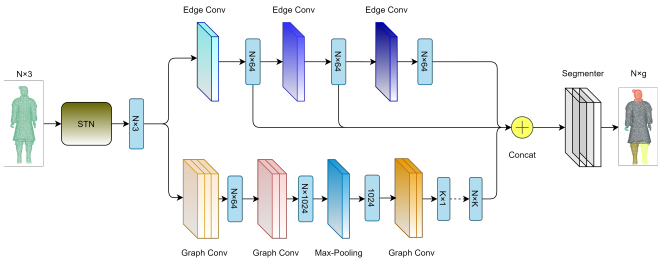

Figure 2: Network Structure. The Yellow Part Is the Bottleneck, Stn Is the Spatial Transform Network, the Part between Stn and Bottleneck Is the Encoder, the Part after the Bottleneck Is the Segmenter.

two points into one cluster. The description of the algorithm is as follows:

1. Input:

(a) xyz coordinates of each point $p_n \in R$ (b) normal value of each point $n_n \in R$ (c) Euclidean distance of every two points $e_n \in R^n$ (d) neighborhood of each point $b_n \in R^k$ 2. The randomly chosen point is added to the set A, we can call it seed a.

3. For other points which are not transformed to seeds in A:

(a) Each of point in seed's neighbor N_a, check the normal distance and Euclidean distance between the point in N_a and the seed a.
(b) We set the normal distance and the Euclidean distance threshold. If both of value is less than the two thresholds, we can add it to A.
(c) The current seed is added to the checked list S.

3.2 SRG-Net

SRG-Net is inspired by the dynamic graph in DG-CNN and auto-encoder in FoldingNet. Unlike classical graph CNN, our graph is dynamic and updated in each layer of the network. Compared with the methods that only focus on the relationship between points, we also propose an encoder structure to better express the features of the entire point cloud, aiming at learning the structure of the point cloud and optimize the pre-segmentation results of SRG. Our network structure can be shown in Figure 2, which consists of two sub-network, the first part is an encoder that generates features from the dynamic graph and the whole point cloud and the second is the segmentation network.

Point cloud is denoted as S. Lower-case letters are used to represent vectors, such as x, and upper-case letters are used to represent matrix, such as A. We call a matrix m-by-n or $m \times n$ if it has m rows and n columns. In addition, the terracotta warrior point cloud data is N points with 6 features $x,\ y,\ z,\ N_x,\ N_y,\ N_z$(xyz coordinates and normal values), denoted as $X = \{x_1,\ x_2,\ x_3 \ldots x_n\} \subseteq R^6$.

3.2.1 Encoder Structure. The SRG-Net encoder follows a similar design of [2], the structure of SRG-Net is shown in Figure 2. Compared with [2], our encoder concatenate several multi-layer perceptrons (MLP) and several dynamic graph-based max-pooling layers. The dynamic graphs are constructed by applying KNN on point clouds. For the entire point cloud S, we compute a spatial transformer network and get a transformer matrix of 3-by-3 to maintain invariance under transformations. Then for the transformed point cloud, we

compute three dynamic graphs and get graph features respectively. In graph feature extracting process, we adopt the Edge Convolution in [2] to compute the graph feature of each layer, which uses an asymmetric edge function in Eq. 1) (see also in Figure 3):

$$f_{ij} = h\left(x_i,\ x_i - x_j\right) \tag{1}$$

where it combines the coordinates of neighborhood center x_i with the subtraction of neighborhood point and the center point coordinates $x_i - x_j$ to get global and local information of neighborhood. Then we define our operation in Eq. 2):

$$g_{ij} = \Theta(\mu \cdot (x_i - x_j) + \omega \cdot x_i) \tag{2}$$

where μ and ω are parameters and Θ is a ReLU function. Eq. 2) is implemented as a shared MLP with Leaky ReLU. Then we define our max-pooling operation in Eq. 3):

$$x_i = \max_{j \in N(i)} g_{ij} \tag{3}$$

where $N(i)$ means neighborhood of point i.
The bottleneck is computed by the graph feature extraction layer. The structure is shown in Figure 2. First, The covariance 3×3 matrix is computed for every point and vectorized to 1×9. Then the $n \times 3$ matrix of point coordinates is concatenated with the $n \times 9$ covariance matrix into a $n \times 12$ matrix. Then we put the matrix into a 3-layer perceptron. Then the output of the perceptron is fed to two subsequent graph layers. In each layer, max-pooling is added to the neighbor of each node. At last, a 3-layer perceptron is applied to the former output and get the final output. The whole process of the graph feature extraction layer is summarized in Eq. 4):

$$Y = I_{max}(X)\, K \tag{4}$$

In Eq. 4), X is the input matrix to the graph layer and K is a feature mapping matrix. $I_{max}(X)$ can be represented in Eq. 5:

$$(Imax(X))\, ij = \Theta\left(\max_{k \in N(i)} x_{kj}\right) \tag{5}$$

where Θ is a ReLU function and $N(i)$ is the neighborhood of point i. The max-pooling operation in Eq. 5) can get local feature based on the graph structure. So the graph feature extraction layer can not only get local neighborhood features, but also global features.

3.2.2 Segmenter Structure. Segmenter gets dynamic graph features and bottleneck as input and assign labels to each point to segment the whole point cloud. First, bottleneck is replicated N times in Eq. 6):

$$B' = B \cup B \underset{N}{\cup} \cdot s \cup B \tag{6}$$

where N is the number of points in point cloud and B is the bottleneck. The output of replication is concatenated with dynamic features in Eq. 7):

$$C = B\prime \cup D_1 \cup D_2 \cup D_3 \tag{7}$$

where D_1, D_2, D_3 represent dynamic graph features. At last,the output of concatenation is fed to a multi-layer-perceptron to segment the point cloud in Eq. 8).

$$D = \Theta(\Psi \cdot C + \Omega) \tag{8}$$

where Ψ and Ω represent parameters in the linear function, and Θ represents a ReLU function.

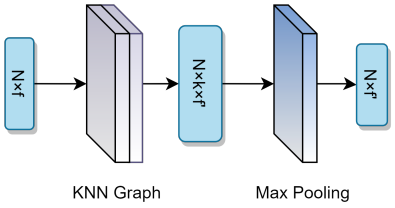

Figure 3: Edge Convolution. The Figure Contains a KNN Graph Convolution Mapping the Point Cloud to N ×k × f', and then Put into Max-Pooling to Get the Point Cloud Edge Feature.

3.3 Refinement

Inspired by [23], point refinement is designed to achieve better segmentation results. The basic concept of point clustering is to group similar points into clusters. In point cloud segmentation, it is intuitive for the clusters of points to be spatially continuous. We add the constraint to favor cluster labels that are the same as those of neighboring points.K superpoints $\{S_k\}_{k=1}^{K} = 1$ are extracted from the input point cloud L , where S_k is a set of the indices of points that belongs to the k-th superpoint. Then points in each superpoint are assigned the same cluster label. More specifically, letting $|c_n|_{n \in S_k}$ be the number of points in S_k that belong to the c_nth cluster, we select the most frequent cluster label c_{max}, where $|c_{max}|_{n \in Sk} \geq |c_n|_{n \in Sk}$ for all c_n *in* 1, ..., q. The cluster labels are replaced by c_{max} for $n \in S_k$.
In this paper, there is no need to set a large number in seed-region-growing process because our neural network can learn both the local and global features. Seed-region-growing method is chosen with K = 25 for the super point extraction. The refinement process can achieve better results compared with straight learning. In addition, instead of using ground truth to calculate loss,
we use the coarse segmentation result of seed-region-growing method to calculate the loss.

4 EXPERIMENTS

4.1 Experimental Protocols

We conduct experiments on terracotta warrior dataset [24] and ShapeNet dataset. We implement the pipeline using PyTorch and Python3.7. All the results are based on experiments under RTX 2080 Ti and i9-9900K. The performances of each method in the experiment are evaluated by the accuracy (mIoU) and the latency. Point clouds are uniformly sampled to 10,000 points thus as the experimental inputting. We select SGD as optimizer, choose lr = 0.005, and set the momentum factor to 0.1. To make fair comparison in the experiments, we combine PointNet [19], PointNet2 [13], DGCNN [3], PointHop++ [25] with the seed region growing method to have a similar structure of SRG-Net. We also set the number of iterations T as 2000.

4.2 Experiments on Terracotta Warrior

We use Artec Eva [24] to collect 500 intact terracotta warrior models, and we take 400 of the 500 models as the training set and 100 as the validation set. Each model consists of about 2 million points which

Table 1: Comparison of Different Methods

Method	Accuracy	Latency
SRG-DGCNN	78.32%	37.26%
SRG-PointHop++	77.94%	30.62%
SRG-ECC	76.32%	31.45%
SRG-PointNet	72.03%	13.36%
SRG-PointNet2	70.55%	24.10%
K-means-DGCNN	70.64%	36.30%
K-means-Pointnet2	52.33%	25.38%
K-means-Pointnet	53.94%	13.47%
SRG-Net	82.63%	22.11%

include xyz coordinates, vertical normals, triangle meshes and RGB data. Before the experiments start, we eliminate the triangle meshes and RGB data of the original models, and remain xyz coordinates and vertical normals. Moreover, we uniformly sample the above point clouds to 10,000 points thus as the experimental inputting. In reality, the terracotta warriors are generally unearthed in the form of limb fragments.

To better assist the restoration of terracotta warriors, we split the point cloud into six pieces both in SRG and K-Means. We also set the number of iterations T as 2000 in each method (PointNet [12], PointNet2 [23], DG-CNN [3], PointHop++ [25], ECC [26]) for every point cloud. Unlike the supervised problem, our unsupervised method solves two problems: prediction of cluster labels with fixed network parameters and training of network with predicted labels. The former sub-problem is solved with Section 3.3. The latter sub-problem is solved with back-propagation.

In addition, the seed region growing is also compared with K-means. Comparison of different methods is shown in Table 1Ranking the results of different methods in Table 1, we can draw the following conclusions:

1. Compared with SRG-DGCNN, SRG-PointNet, SRG-PointNet2, our network can enhance the accuracy of learning with less latency.
2. Compared with Kmeans-DGCNN, Kmeans-PointNet, Kmeans-PointNet2, our customized SRG method can achieve better results compared with k-means.
3. In general, SRG-Net has obvious advantages in accuracy and latency.

4.3 Experiments on ShapeNet

In this section, we perform experiments on ShapeNet to show the robustness of our SRG-Net method. ShapeNet contains 16,881 objects from 16 categories. We set the number of the parts according to different categories separately. All models were down-sampled to 2,048 points. The evaluation metric is mean intersection-over-union (mIoU).

Some of our quantitative experiment results are shown in Table 2. As in Table 2, SRG-Net outperforms all previous models. SRG-Net improves the overall accuracy of SRG-DGCNN by 8.2% and even larger overhead compared with SRG-PointNet2 and SRG-PointNet. Especially, our method outperforms SRG-DGCNN on all kinds of categories, increasing 5% accuracy on knife. Overall, our method achieves better accuracy on ShapeNet compared with other methods.

Figure 4: SRG-Net Segmentation Results of ShapeNet.

Some visualization results are shown in Figure 4. As is shown in Figure 4, SRG-Net achieves good results on bag, knife, motorbike, and achieves quite good results on airplane, earphone and laptop.

4.4 Ablation Study

In order to show the influence of different modules in our method, we conduct ablation studies on our terracotta warrior dataset, which are described in Section 3.2, which are evaluated by overall accuracy and mIoU.

The influences of different modules. As shown in Table 3, The results of version without the graph convolution (A, row 1) show that the network not working well in learning the topological features of the local neighborhood of the point cloud. The results of version without edge convolution (B, row 2) demonstrate that our method without edge convolution will cause the network unable to understand the relationship between points well. The results of version without refinement operation (C, row 3) reveal that the pipeline cannot set tags reasonably based on point cloud content because the number of unique cluster labels should be adaptive to context. The results show that the refinement operation increase 15.6% on the accuracy of SRG-Net.

5 CONCLUSION

This paper provides an end-to-end model called SRG-Net for unsupervised segmentation on the terracotta warrior point cloud. Our method aims at assisting the archaeologists in speeding up the repairing process of terracotta warriors. Firstly, we design a novel seed-region-growing method to pre-segment point clouds with coordinates and normal value of the 3D terracotta warrior point clouds. Based on the pre-segmentation process, we proposed a CNN inspired by the dynamic graph in DG-CNN and auto-encoder in FoldingNet to learn the local and global characteristics of the 3D point cloud better. At last, we propose a refinement process and append it after the CNN process. Finally, we evaluate our method on

Table 2: Accuracy (%) of Different Methods on ShapeNet

Method	Airplane	Bag	Earphone	Knife	Laptop	Bike	Mug	OA
SRG-DGCNN	67.7393	61.2568	56.0633	77.0035	75.8854	70.7194	59.3024	72.6273
SRG-Pointnet2	61.1865	54.4515	51.6673	71.9515	69.9074	64.3308	53.8897	66.4712
SRG-Pointnet	57.1590	51.0517	47.2101	67.9229	66.4234	61.2438	49.1265	61.4242
SRG-Net	73.0493	67.0935	62.4256	83.2430	82.6076	76.7677	65.8810	78.1954

Table 3: Ablation Study

Method	**mIoU(%)**	**OA(%)**
Without Graph Conv	77.71	80.17
Without Edge Conv	74.18	76.51
Without Refinement	71.46	73.48
Ground Truth	81.72	82.63

the terracotta warrior data, and we outperform the state-of-the-art methods with higher accuracy and less latency. Besides, we also conduct experiments on the ShapeNet, which demonstrates our approach is also useful for the human body and standard object part segmentation. Our work still has some limitations. The number of points from the hand of the terracotta warrior is relatively small, so it is difficult for SRG-Net to learn the characteristics of hands. We will try to work on this problem in the future. We hope our work can be helpful to the research of terracotta warriors in archaeology and point cloud work of other researchers.

ACKNOWLEDGMENTS

This work is equally and mainly supported by the National Key Research and Development Program of China (No. 2019YFC1521102, No. 2019YFC1521103) and the Key Research and Development Program of Shaanxi Province (No. 2020KW-068). Besides, this work is also partly supported by the Key Research and Development Program of Shaanxi Province (No.2019GY-215, No.2019ZDLSF07-02, No.2019ZDLGY10-01), the Major research and development project of Qinghai (No. 2020-SF-143), the National Natural Science Foundation of China under Grant (No. 61701403, No. 61731015) and the Young Talent Support Program of the Shaanxi Association for Science and Technology under Grant (No.20190107).

REFERENCES

[1] A. Tao (2020). Unsupervised point cloud reconstruction for classific feature learning. https://github.com/AnTao97/Unsupervised.
[2] Y. Yang, C. Feng, Y. Shen, and D. Tian (2018). "Foldingnet," in *CVPR,* (2018 June, Salt Lake City), pp. 206-215.
[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon (2019). Dynamic graph cnn for learning on point clouds. Acm Transactions On Graph. (tog) **38**: 1-12.
[4] F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg (2017). Deep projective 3d semantic segmentation, in *CAIP,* (2017 August , Springer, Ystad), pp. 95-107.
[5] A. Boulch, B. Le Saux, and N. Audebert (2017). Unstructured point cloud semantic labeling using deep segmentation networks. 3DOR **2**: 7.
[6] B. Wu, A. Wan, X. Yue, and K. Keutzer (2018). "Squeezeseg," in *2018 IEEE International Conference on Robotics and Automation (ICRA),* (2018 May, IEEE, Brisbane), pp. 1887-1893.
[7] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss (2019). "Rangenet++," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* (IEEE), pp. 4213–4220.
[8] B. Graham, M. Engelcke, and L. Van Der Maaten (2018). 3d semantic segmentation with submanifold sparse convolutional networks, in *Proceedings of the IEEE conference on computer vision and pattern recognition,* (2018 June, Salt Lake City), pp. 9224-9232.
[9] A. Dai and M. Nießner (2018). "3dmv," in *ECCV,* (2018 September, Munich), pp. 452-468.
[10] M. Jaritz, J. Gu, and H. Su (2019). "Multi-view pointnet for 3d scene understanding," in *Proceedings of the IEEE International Conference on Computer Vision Workshops,* (2019 October, Seoul), pp. 0-0.
[11] S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun (2018). "Deep parametric continuous convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* (2018 June, Salt Lake City), pp. 2589-2597.
[12] C. R. Qi, H. Su, K. Mo, and L. J. Guibas (2017). "Pointnet," in *CVPR,* (2017 July, Hawaii), pp. 652-660.
[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas (2018). "Pointnet++," in *Advances in neural information processing systems,* (2018 December, Montreal), pp. 5099–5108.
[14] M. Jiang, Y. Wu, T. Zhao, Z. Zhao, and C. Lu (2018). "Pointsift," arXiv preprint arXiv:1807.00652.
[15] H. Zhao, L. Jiang, C.-W. Fu, and J. Jia (2019). "Pointweb," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* (2019 July, Long Beach), pp. 5565-5573.
[16] W. Wu, Z. Qi, and L. Fuxin (2019). "Pointconv," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* (2019 July, Long Beach), pp. 9621-9630.
[17] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen (2018). "Pointcnn: Convolution on X-transformed points," (2018 June, Salt Lake City).
[18] Z. Wang and F. Lu (2019). "Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes," IEEE transactions on visualization computer graphics.
[19] D. OuYang and H.-Y. Feng (2005). "On the normal vector estimation for point cloud data from smooth surfaces," Comput. Des. **37**: 1071-1079.
[20] J. Zhou, H. Huang, B. Liu, and X. Liu (2020). "Normal estimation for 3d point clouds via local plane constraint and multi-scale selection," Comput. Des. **129**, 102916.
[21] Y. Wang, H.-Y. Feng, F.-É. Delorme, and S. Engin (2013). An adaptive normal estimation method for scanned point clouds with sharp features. Comput. Des. **45**, 1333–1348.
[22] R. B. Rusu and S. Cousins (2011). "3d is here," in *2011 IEEE international conference on robotics and automation,* (IEEE, 2011 May, Shanghai), pp. 1-4.
[23] A. Kanezaki (2018). Unsupervised image segmentation by backpropagation," in *ICASSP,* (IEEE, 2018 April, Calgary), pp. 1543-1547.
[24] A. Europe, "3d object scanner artec eva".
[25] M. Zhang, Y. Wang, P. Kadam, S. Liu, and C.-C. J. Kuo (2020). "Pointhop++" arXiv preprint arXiv:2002.03281.
[26] M. Simonovsky and N. Komodakis (2017). "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *CVPR,* (2017 July, Hawaii).