3D Object Recognition Method Based on Point Cloud Sequential Coding

Shuai Dong, Li Ren, Kun Zou and Wensheng Li*
University of Electronic Science and Technology of China, Zhongshan Institute
No. 1 Xueyuan Road, Zhongshan
China
0760-88227202, 086
dongshuai@zsc.edu.cn; cosmrya@gmail.com; cszoukun@foxmail.com; Lws7166@126.com

ABSTRACT
Point cloud is the most usual manner to describe 3-dimensional (3D) objects. It contains the position of each point of the object in spatial rectangular coordinate system and their corresponding RGB pixel values, which can describe the object appearance completely. However, point cloud data is dense and unordered, and hence not suitable for convolutional neural networks (CNNs) that is the basic of deep learning. In this work, a new 3D object recognition method based on sequential coding of point cloud is proposed, with which a point cloud can be transformed into an ordered multi-channel 2D array that is suitable for efficient 2D convolutional operation. Point cloud can be sequential coded in spherical coordinate or cylindrical coordinate. Based on the new ordered data, a classifying network and a retrieval framework is design for recognition of 3D objects. The proposed method has achieved better results in both classification and retrieval tasks than other methods.

CCS Concepts
•**Computing methodologies→Object recognition**.

Keywords
Point cloud; Convolutional neural networks; Point cloud ordered coding; 3D object recognition

1. INTRODUCTION
All objects in the real world are three-dimensional (3D) and have multiple views. One object may looks very different at different position, and two objects may look same from some specific angles of view. That is because a single view cannot completely describe the appearance information of an object, and classification or retrieval task based only on a single view would lead to poor results. For recognition of 3D objects, we must make full use of the structural topological and color information, that is the appearance information of objects.

ICMLC 2020, February 15–17, 2020, Shenzhen, China

ACM ISBN 978-1-4503-7642-6/20/02…$15.00
DOI: https://doi.org/10.1145/3383972.3383984

The presentation methods of 3D objects based on voxel or point cloud can describe appearance information completely, and based on them VoxelNet and PointNet [5] are proposed. However, both of them are suffering from large amount of computation and difficulty in training. To avoid this problem, a new 3D object recognition method based on sequential coding of point cloud is proposed, with which point cloud can be transformed into an order multi-channel 2D array that is suitable for efficient 2-dimension (2D) convolutional operation. Point cloud can be coded in both spherical coordinates and cylindrical coordinates. The proposed method has achieved better results on both classification and retrieval tasks of 3D objects than other methods.

2. RELATED WORK
Objects in reality exist in a stereoscopic form, while a 2D view is only a projection of an object from a certain perspective. Because one single 2D view can't describe appearance information of 3D objects completely, those recognition methods based on 2D view can't achieve satisfactory accuracy.

Since there exists this limitation for 2D-view-based methods, many researchers began to study 3D-representation-based technologies. Knopp etc. [5] use a SIFT and SURF feature descriptor that can be extended into a 3D voxel grid to represent the appearance feature of one object. Chaudhuri etc. [6] use a local shape diameter measured at densely sampled surface points to represent a 3D object. Kokkinos etc. [4] use thermonuclear features on a polygon mesh to represent a 3D object. These methods depends on artificially designed 3D features based on the geometric properties of the surface or structure of the object. However, they are not robust enough. Thanks to the development of deep learning, more and more researchers have begun to explore 3D object recognition methods based on deep learning and convolutional neural networks (CNNs), and developed varieties of deep neural networks based on spatial presentation methods or multi-view representation methods. Wu etc. [8] propose a three-dimensional object recognition method based on voxel model and 3D convolution. Compared to 2D data and 2D convolution, 3D manner has higher spatial and temporal complexity, while lower resolution may lead to much worse results. Qi etc. [1] proposes a network structure called PointNet. This network uses a spatial transformation network to transform the input point cloud data to extracted features successively, which solves the rotation problem of the point cloud to some extent. It uses the maximum pooling operation to extract the global features of the point cloud data, solving the problem of unordered. But PointNet only considers global features and would lose local information between adjacent points. To this end, PointNet++ [2] is proposed. Its basic idea is to select some important points as the center point of each local area, then select

several neighbor points around each of them. These neighbor points are treated as a local point cloud and features can be extracted from them with PointNet. All these researches are based on 3D representation methods, and are suffering from large amount of computation. Su etc. [3] proposes a new method multi-view convolution networks (MVCNN) based on multi-view representation method. The input of the network is the stack of several ordered 2D views, and the view pooling layer is used to fuse multiple view features. MVCNN is better than those 3D representation methods in both classification and retrieval tasks. However, it is necessary to ensure that the input 2D views can contain complete appearance information. It means that the number of views must be enough. For simple object, such as a book or a bowl, 2 views are enough. But when the appearance is complex, maybe more than 8 views is necessary to achieve high accuracy.

3. METHOD

3.1 Sequential Coding for Point Cloud

There are many methods to make point cloud ordered. For example, if unordered point cloud data (x, y, z, r, g, b) are sorted according to xyz coordinates and sampled in voxel grids, it becomes ordered and suitable for 3D convolution [7] operation. However, due to relatively dense data points, 3D convolution requires a large amount of computation and high performance hardware. In this section, sequential coding method for cloud point, which uses trigonometric transformation to convert point cloud data into cylindrical coordinates(ρ, ϕ, z, r, g, b) or spherical coordinates $(\rho, \theta, \phi, r, g, b)$, is discussed. According to the interval of z and ϕ or θ and ϕ after sorting and zoning, the points can be ordered in the form of 3D matrix, which is sparser than voxel gird, and then the computation is less than VoxelNet.

3.1.1 Spherical coordinate ordering

To describe point cloud in spherical coordinate, we need to transfer point cloud P from the rectangular space coordinate system D into a spherical coordinate system S, that is transform $p = (x_k, y_k, z_k, r_k, g_k, b_k)$ into $p^* = (\rho_h, \theta_h, \phi_h, r_h, g_h, b_h)$, where $p \in D, k \in |D|$ and $p^* \in S, h \in |S|$. The coordinates of a point in space rectangular coordinate system (x, y, z) and its corresponding point (ρ, θ, ϕ) in spherical coordinate system has the transformation relationship as shown in Figure 1.

After the spherical coordinate p^* of point cloud data is obtained, θ ϕ will be uniformly divided in [0,360) and [0,180) to generate matrix indices according to the steps of discretization. For example, if step is set to 2, the sequence of θ is $l_1 =$ [0,2,4,6,8,…,180) and ϕ is $l_2 =$ [0,2,4,6,8,…,360). And then, in order to establish the ordered matrix $\boldsymbol{M}_t$ along the longitudinal axis and the horizontal axis, elements $m_{\theta_i \phi_j} = (\rho_h, r_h, g_h, b_h)$ are divided into range of the sampling points respectively by the two sequences, where $\theta_i \in l_1, \phi_j \in l_2$, integer $i \in [0,90)$ and $j \in [0,180)$. So the new matrix $\boldsymbol{M}_t$ is obtained as a sparse four-channel 2D array. For the same θ_i and ϕ_j, points in this range can be divided from the surface of the object into the different matrix $\boldsymbol{M}_t, t \in |$ ray l through the surfaces|, as Figure 2 depicted. Since it is enough to provide complete appearance information of one object, only $\boldsymbol{M}_1$ and $\boldsymbol{M}_4$ are selected here. It means that the final ordered matrix M is a sparse eight-channel 2D array.

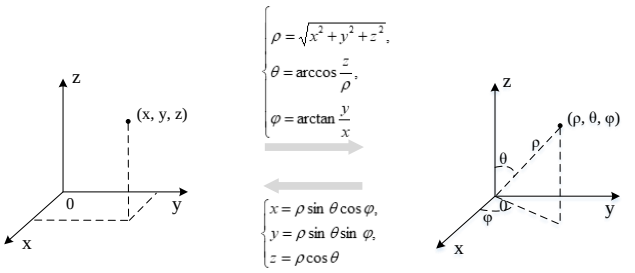

Figure 1. Transformation relationship between space rectangular coordinates and spherical coordinates

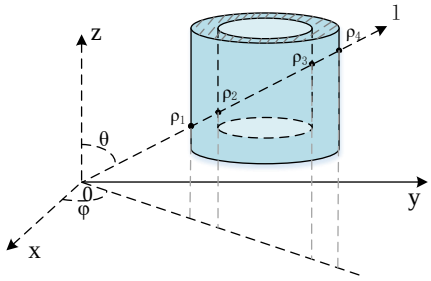

Figure 2. Points of an object in spherical coordinates

3.1.2 Cylindrical coordinate ordering

To describe unordered point cloud cylindrical coordinates, first of all we need to transform point cloud P from the space rectangular coordinate D into a cylindrical coordinate system C, that is the point $p = (x_k, y_k, z_k, r_k, g_k, b_k)$ into $p^* = (\rho_h, \phi_h, z_h, r_h, g_h, b_h)$, where $p \in D, k \in |D|$ and $p^* \in C, h \in |C|$. The coordinates of a point in space rectangular coordinate system (x, y, z) and to its corresponding point (ρ, ϕ, z) in cylindrical coordinate system has the transformation relationship depicted in Figure 3.

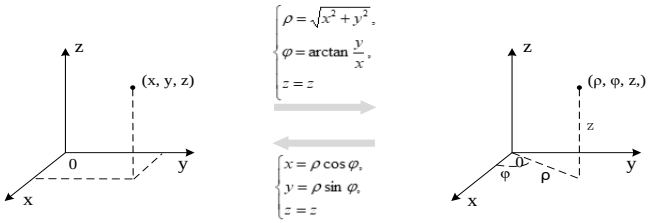

Figure 3. Transformation from rectangular coordinates to cylindrical coordinates

Denote the point cloud in cylindrical coordinates as p^*. Divide the z into 90 interval between the minimum and maximum values, then we can get the value sequence $l_1 = \left[min, \dots, min + 89 * \frac{max - min}{90max}\right)$, where ϕ ranges in $[0,2\pi)$ according to the angle of the size of the discretization. For example, if step is 2, we would get the sequence $l_2 = [0,2,4,6, \dots, 2\pi)$. Then, with the two sequences as indices, we can establish the order matrix $\boldsymbol{M}_t$ along longitudinal axis and horizontal axis. Matrix element $m_{z_i \phi_j} = (\rho_h, r_h, g_h, b_h)$ is divided into ranges that are determined by the sampling points, including $z_i \in l_1, \phi_j \in l_2$,integer $i \in [0,90)$ and $j \in [0,180)$. A new sparse matrix $\boldsymbol{M_t}$ which is a four-channel two-dimensional array is obtained. For the same z_i and ϕ_j, points can be divided into different groups on the surface of the object according to different and ρ value, which are corresponding to the

different matrices M_t, $t \in |$ ray l through the surfaces$|$, as shown in Figure 4. Since it is enough to provide complete appearance information of one object, only $\boldsymbol{M_1}$ and $\boldsymbol{M_4}$ are selected here. It means that the final ordered matrix $\boldsymbol{M}$ is a sparse eight-channel 2D array.

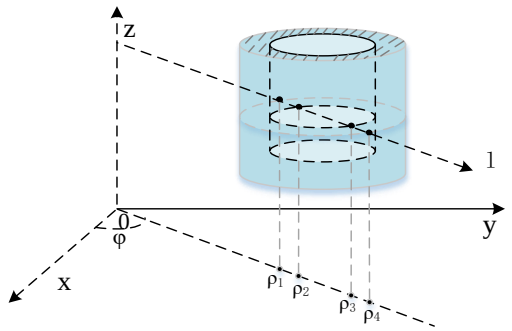

Figure 4. Points of an object in cylindrical coordinates

3.2 Structure of the Networks

After recoding the point cloud with those two ways in section 3.1, a sparse and orderly eight-channel 2D array can be obtained, on which efficient 2D convolution operation can be directly used. To validate the effectiveness of the proposed method in 3D object classification tasks, a convolutional network is designed. Its structure is shown in Figure 5. The body of the network contains five CNNs layers, which are used to exact the spatial and color features. And the head contains 3 full-connection (FC) layers, which is used to make decisions.

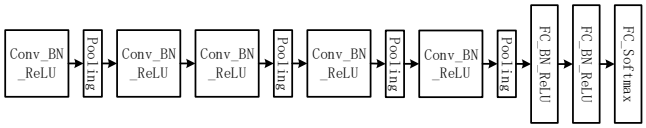

Figure 5. Convolutional network structure

After a series of related operations via the network above, the input data can predict the classifying score at the last layer, and the output of the last CNNs layer can be used as the feature vector of the input data. Denote the input as X, and $f()$ as the mapping relationship between the input data and the classifying score of the model; denote s as the category score, then we have $s = f(X)$. The purpose of training is to search the optimal approximation of $f()$ to accurately predict the category of input ordered point cloud. To improve the identification of features, a regular loss function as follow is used:

$$L_{total} = L(\boldsymbol{s}, \hat{\boldsymbol{s}}) + \lambda \|w\|^2 \#(1)$$

In formula (1), $L(\boldsymbol{s}, \hat{\boldsymbol{s}}) = \frac{1}{m}\sum_{i=1}^{m} -\hat{\boldsymbol{s}}^{(i)} \, log \, \boldsymbol{s}^{(i)}$ it the cross entropy loss between the category label $\hat{\boldsymbol{s}}$ and the prediction score s, where m is the number of samples in a batch. Item $\|w\|^2 = \frac{1}{2}\sum_{w_i \in W} w_i^2$ is the L2 regularization, where w is the parameter of the model, and λ is the penalty coefficient that is set to $5*10^{-4}$ here.

3.3 Object Retrieval

3.3.1 Similarity matching model

To finish retrieval tasks, a matching network is designed first. The structure is shown in Figure 6. It can judge whether a 2D features and a 3D features are from the same object. Denote the 2D features and 3D features as $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ respectively, then the input of the matching network is $u = (\boldsymbol{v_1}, \boldsymbol{v_2})$. The loss function of this model adopts the same form as equation (1). The output is the probability value that $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ from the same object. 0 is from different objects, and 1 is from one object. The structure is similar with the head of classifying network except the input vector.

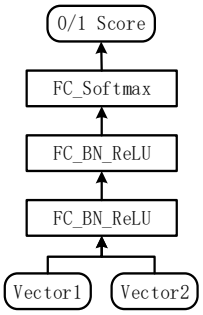

Figure 6. Similarity matching model

3.3.2 Prediction process

The complete retrieval process of 3D object based on sequential coding point cloud is depicted in Figure 7. The process can be divided into two parts: 1) the construction of sample feature database and 2) similarity matching. In the first part, point cloud data of objects are collected, and an ordered eight-channel 2D array for each object is obtained with sequential coding in spherical or cylindrical coordinates. Then, the array is processed through the CNNs in Section 3.2 to obtain an eigenvector that is the named 3D feature and contains complete 3D spatial and color appearance information of the object. Finally, the 3D feature is registered in the database. In the second part, the ordinary 2D feature of a newly object is exacted from one single view with general CNNs. This new 2D feature and each 3D feature in the database constitute an input pair for the match network. We will get a matching score for each object in the database.

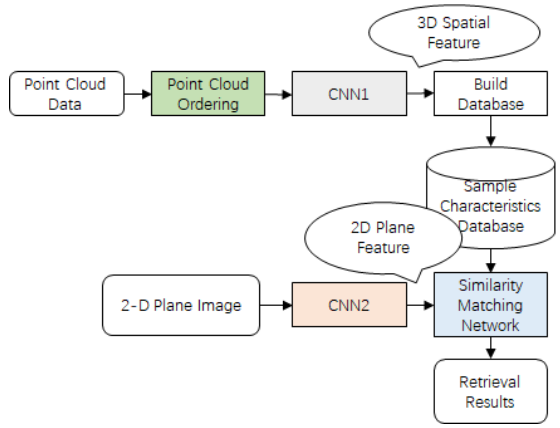

Figure 7. Retrieval process based on point cloud data

4. SIMULATION EXPERIMENT

4.1 Data Sets

The point cloud data and 2D view data used in this paper were generated with 3DsMax and Unity 3D. We established a dataset that contain totally 95 3D object models of different categories. Each object includes 100 groups of point clouds which are have different rotation and displacement and 100 2D views projected from different perspectives. All datasets were divided into training

set, verification set and test set according to the ratio of 6:1:3 in all training procedures.

4.2 The Experimental Setup

In order to verify the effectiveness of 3D object recognition method based on point cloud sequential coding, three groups of experiments were set up: 1) 3D object retrieval based on ordered spherical coordinates; 2) 3D object retrieval based on ordered cylindrical coordinates; 3) 3D object retrieval based on single view features.

Multi-stage model training is required for both groups of experiments. In order to properly train and evaluate the model, data sets required for each stage are allocated in the following ways: (1) The training set of 2D view is used to training a general network which extracts 2850 (95*30) single-view features from the test set. (2) The classifying network shown in Figure 5 is trained with sequential coded training set in different coordinate. We can obtain 2850 (95 * 30) 3D features from the test set with the trained classifying network. Through pairing 3D features and 2D features randomly, 85500 positive samples (3D features and 2D features from the same object) and 89300 negative samples (3D features and 2D features from different objects) are generated. Those positive and negative samples are used to train the matching network shown in Figure 6.

The same training method was used in all stages of those three groups of experiments. Adam acceleration algorithm and bath normalization were adopted, and batch size was set to 50. The learning rate was set to 10^{-2} at beginning and then decrease to 10^{-3} and 10^{-4} after stabilization, momentum was set to 0.9 and dropout rate was set to 0.3.

4.3 Experimental Results and Analysis

Table 1. Experimental results

Test	Classify task	Retrieval task	Discrimination degree		
	acc/%	acc/%	*top2_dst*	*top1_dst*	$\log\frac{top2\_dst}{top1\_dst}$
1	97.123	78.386	0.15512	0.00974	1.20211
2	96.070	79.564	0.17443	0.00741	1.37180
3	80.737	64.197	0.14976	0.01404	1.02803

Note: Test 3 is single view retrieval.

Table 1 shows the results of those 3 groups of experiments. From the table, we can find that the experimental results of Experiment 1) and Experiment 2) are very close, and both of them are significantly better than single view retrieval in classifying and retrieval tasks. It proves the effectiveness of the method. And by using 2D convolution after sequential coding, the computation complexity is greatly reduced, and the training speed is increased greatly. However, during the sequential coding process, there are still lost some 3D appearance information lost. That's because the interval on sphere or cylinder cannot be evenly divided and uniformly sampled. For example, those points close the spherical poles in spherical coordinate are sparse and points one equator are dense. Therefore, the 3D information of the object described by the ordered sparse matrix is uneven in Cartesian space. For the cylindrical coordinates after conversion, uniform sampling can be performed. However, if the object information is concentrated on the upper and lower surfaces of the cylinder, interval division and sampling will also lose a lot of important information.

5. CONCLUSION

In this paper, a 3D object recognition method based on point cloud sequential coding is proposed. Two methods to recoding unordered point cloud in spherical coordinates and cylindrical coordinates are studied. The unordered and dense point cloud data can be converted into ordered and sparse multi-channel 2D array which is suitable for efficient 2D convolution operation. And experimental results show that this method can achieve excellent accuracy in both classification and retrieval tasks of 3D objects.

6. ACKNOWLEDGMENTS

This work is supported by Research Foundation of University of Electronic Science and Technology of China, Zhongshan Institute, under Grant 417YKQ12, 419YIY04 and 419N26.

7. REFERENCES

[1] C. R. Qi, H. Su, K. Mo, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 2017, 77-85.

[2] C. R. Qi, L. Yi, H. Su, et al. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space[C]. *Proceedings of the Annual Conference on Neural Information Processing Systems*, Long Beach, 2017, 5105-5114.

[3] H. Su, S. Maji, E. Kalogerakis, et al. Multi-view Convolutional Neural Networks for 3D Shape Recognition[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Santiago, 2015, 945-953.

[4] I. Kokkinos, M. M. Bronstein, R. Litman, et al. Intrinsic shape context descriptors for deformable shapes[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Providence, 2012, 159-166.

[5] J. Knopp, M. Prasad, G. Willems, et al. Hough Transform and 3D SURF for Robust Three Dimensional Classification[C]. *Proceedings of the European Conference on Computer Vision*, Heraklion, 2010, 589-602.

[6] S. Chaudhuri, V. Koltun. Data-driven suggestions for creativity support in 3D modeling[J]. *ACM Transactions on Graphics (TOG)*, 2010, 29(6): 183.

[7] S. Ji, W. Xu, M. Yang, et al. 3D Convolutional Neural Networks for Human Action Recognition[C]. *Proceedings of the 27th International Conference on Machine Learning*, Haifa, 2010, 495-502.

[8] Z. Wu, S. Song, A. Khosla, et al. 3D ShapeNets: A Deep Representation for Volumetric Shapes[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 2015, 1912-1920.