CSCI 5210 Advanced Topics in Computer Graphics and Visualization

Assignment: 3D Point Cloud Retrieval

Due Time: 23:59, 15 Feb. 2019

Late penalty: 5% of full mark per day

1. Introduction

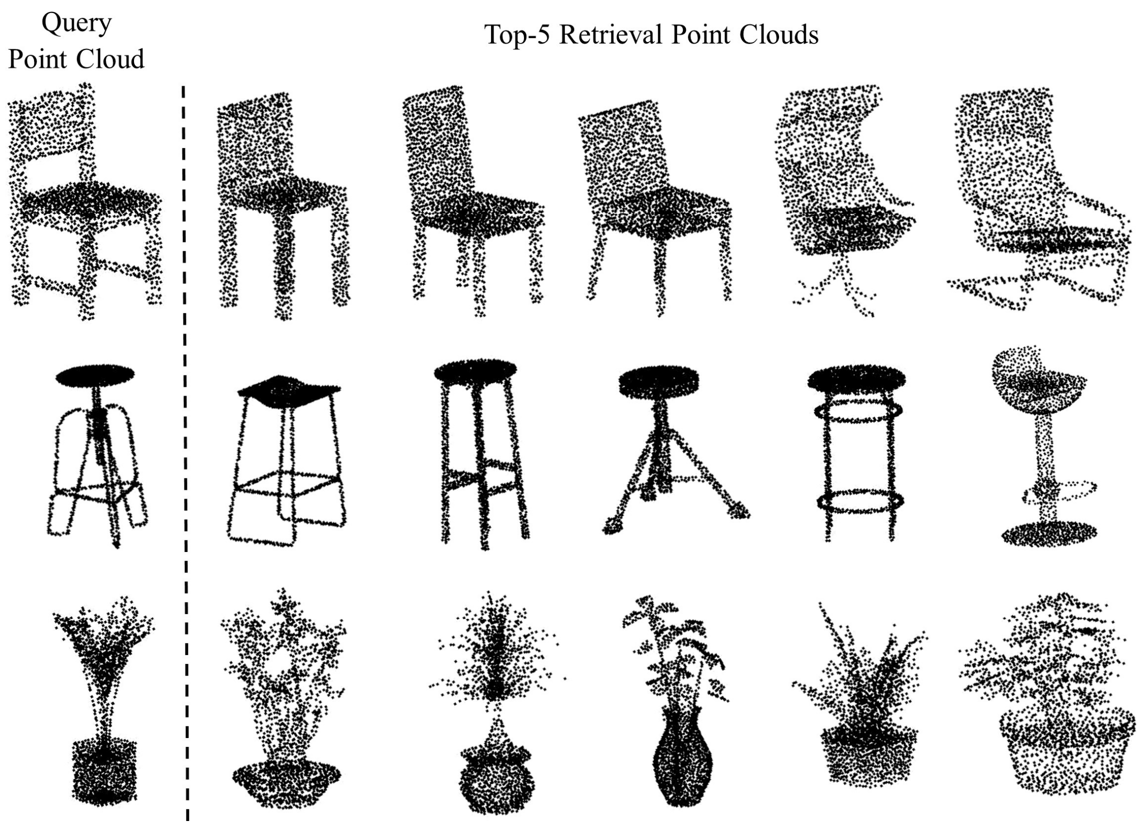

Figure 1: 3D point cloud retrieval results. From top to bottom rows, we show top-5 retrieval results of chair, desk and plant queries.

This assignment will introduce to you an automatic 3D point cloud retrieval technique based on deep learning. In this assignment, you are asked to implement a deep neural network model to learn global features in point clouds and then to search for the objects in a database that are the most similar to a given query object by comparing the similarity of features; see Figure 1 for three examples of the object retrieval results. The objective of this assignment is to learn the basic deep learning technology to process 3D point clouds, which is a recent hot topic in graphics and vision research.

In this assignment, we use "*PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*" [1] as the basic reference. You have to read it carefully before doing this assignment and implement the basic network architecture in this work; details will be provided to you in the following sections in this assignment specification. Further, you can choose to implement your assignment using any deep learning framework, e.g., PyTorch, Caffe, TensorFlow, etc. In the end, you should submit a *report* and *source code* after you finish this assignment. Note also that plagiarism is a serious matter; we will check your code against the code by the other classmates and in the Internet.

2. Implementation Details (75%)

To retrieve geometrically-similar objects, the common procedure is to first compute representative global features for the query object and also for each object in the database; then, we can retrieve similar objects by using nearest neighbor search in the feature space. Next, we will give you the implementation details of each step, as well as the introduction of our provided data.

2.1 Global feature extraction

To learn a global shape signature for an input point cloud, we can train a supervised classification network and use the output of the layer before the score prediction layer as the global feature. To be more specific, the basic part of this assignment is to implement the neural network below. This is a simplified version of the model presented in [1].

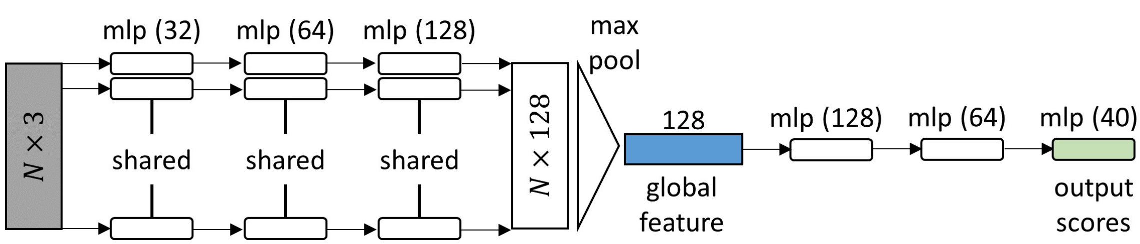

Figure 2: The framework that we will use (N=2048).

The input of the network (the gray box in the above figure) is a point cloud of N points in 3D space. In our provided data, N=2048. For each point, it has (x, y, z) coordinates. The output (the green box) is the prediction scores. Here, we train the network to classify training samples into 40 classes. The blue box indicates the global feature of the input point cloud, which will be used in the next step for nearest neighbor search. In each layer, the number of feature channels is shown in Figure 2 above. Other hyperparameters are listed below:

- Training epoch = 250
- Optimizer: Adam
- Learning rate = 0.001
- Activation function: ReLU

Note that you are encouraged to find other hyperparameters and the number of feature channels in each layer to achieve better performance. You can also adjust the number of convolution layers.

2.2 Nearest neighbor search

The next step is to find the K most similar point clouds in a database given a query point cloud. The idea is very simple. With the above well-trained network, for every point cloud in the database, we can obtain a global feature to represent it. Similarly, for every query point cloud, we can also obtain the global feature. Then, we can simply model the feature distance (e.g., Euclidean distance) as the object similarity. In this way, we could find the K most similar point clouds for each query point cloud. In this assignment, you need to retrieve the top-5 similar point clouds (K=5). The pseudo-code is summarized below for your reference:

Input: a well-trained network, a query point cloud and a searching database
1: compute the global feature of the query point cloud using a well-trained network 2: **for** each point cloud in the database **do** 3:   compute the global feature 4:   calculate the feature distance as the similarity 5: **end for** 6: sort and find the K most similar point clouds
Output: the K most similar point clouds for the given query point cloud

2.3 Provided data

In this assignment, we provide you all the data you need, including the training data, the query point clouds, and the searching database. You can download all the three kinds of data from the course webpage on blackboard. Next, we will briefly introduce each of them.

- Training data

You can use the following code to read the downloaded training data (named *train_data.h5*).

```
import h5py
f = h5py.File('train_data.h5')
data_train = f['data'][:]       ### [9840, 2048, 3]
label_train = f['label'][:]     ### [9840, 1]; This is the ground truth label
```

The provided training data has 9840 point clouds, and there are 2048 points in each point cloud. These point clouds can be divided into 40 classes. You can find the name of each class in "*shape_names.txt.*" Note that the label of each class corresponds to the order of the names in the txt file. For example, the first line in "*shape_names.txt"* is "airplane," so the label of the airplane class is 0; the second line in "*shape_names.txt"* is "bathtub," so the label of the bathtub class is 1; and so on.

- Query data

You can download the query point clouds (named *query_data.h5*) also from the course webpage on blackboard. It contains totally 40 objects for each of the 40 classes. You can check the ground truth label of each query point cloud using **f['label'][:]**.

- Searching database

You can download the searching database (named *database.h5*) from the course webpage. The database contains 800 point clouds from 40 classes. So in each class, there are 20 objects. You can check the ground truth label of each point cloud using **f['label'][:]**.

Your task is to retrieve the five most similar point clouds from the 800 point clouds for each query point cloud. Since we provide you the ground truth labels of each point cloud, you therefore can calculate the accuracy of your retrieval results yourself. The accuracy is defined as the number of retrieval results with correct labels divided by five. For example, assuming that the query point cloud is an airplane, if the retrieved 5 point clouds returned from your program are all airplanes, the accuracy is 100%; if only 4 out of 5 results are airplanes, the accuracy is 4/5=80%.

3. Advanced Topics (25%)

After finishing the above task, we believe that you have the basic ideas of processing point clouds using deep learning techniques. However, since the network architecture shown in Figure 2 is a very simple and basic one, so there will be many retrieval results that are in wrong classes. Therefore, in the advanced part of the assignment, we have to extend your work in whatever aspects to improve the retrieval results. Here are some suggestions that you can consider to improve your current results:

1) Adopt a more powerful network architecture for feature embedding. In Figure 2, we implement the network based on [1]. However, there are more advanced neural network architectures, e.g., PointNet++ [2], PointCNN [3], SPLATNet [4], etc. You can explore these networks (or even design and develop own network) to extract global feature.
2) Replace Euclidean metric with a more powerful metric to calculate the feature distance.
3) Unsupervised point cloud retrieval. In Figure 2, we train the network to learn global features in a supervised manner. However, you may also adopt an unsupervised network to learn the global features. Here, we point to you a reference paper, which is "*FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformance*" [5], which may inspire you.
4) You may even try your method on more comprehensive datasets; if you have good results, you may submit a paper to research conferences such as ICCV or SIGGRAPH Asia.

4. Grading Scheme

Your assignment will be graded by the following marking scheme: basic model [75%] + advanced topics [25%], in these two parts:

- Report (writing, readability) [50%]
  - Should not exceed 4 pages, and use fonts that are no less than 10 in size.
  - Describe the overall structure of your method and briefly describe each step in your method.
  - Show the retrieval accuracy on the query point clouds. If you train your network in a supervised manner, the higher the retrieval accuracy is, the higher your score will be. If you further implement the unsupervised training network, you could get the extra 25 scores.
  - Show some representative visual results, just like Figure 1.
  - Give a brief discussion on the retrieval results.

- Code (readability and correctness) [50%]
  - The network structure [20%]
  - The codes for training [20%]
  - The codes for nearest neighbor search [10%]

Note: zero marks will be given to you (and we have to report to the faculty of engineering), if you download codes from the Internet directly or copy code from other students.

5. Guidelines to Submit Files

1) You are suggested to write your programs on Ubuntu and use the GPU to train the deep models.
2) The list of submission files:
   a. Code (only include the core source code files such as the network structure and the source code files for training and retrieval; do not submit the well-trained model.)
   b. Report (The report should not exceed 4 pages.)
3) Zip all the files in a .zip or .rar file. Name it with your own name and the student ID (e.g. LIXianzhi_115508XXXX.zip), and submit it through the assignment 1 submission box in the course webpage on https://blackboard.cuhk.edu.hk.
4) In case of multiple submissions, only the latest one will be considered.

References

[1] Qi C R, Su H, Mo K, et al. PointNet: Deep learning on point sets for 3D classification and segmentation. CVPR, 2017.

[2] Qi C R, Yi L, Su H, et al. PointNet++: Deep hierarchical feature learning on point sets in a metric space. NIPS. 2017.

[3] Li Y, Bu R, Sun M, et al. PointCNN. arXiv preprint arXiv:1801.07791, 2018.

[4] Su H, Jampani V, Sun D, et al. SPLATNet: Sparse lattice networks for point cloud processing. CVPR, 2018.

[5] Yang Y, Feng C, Shen Y, et al. FoldingNet: Point cloud auto-encoder via deep grid deformation. CVPR, 2018.