

UTBM P2020

UN56 Digital Technologies and Methods in Industry 4.0

Point cloud segmentation (clustering)

Projet 2.1



Application of multiple clustering algorithms

Zihan PENG

Lili CHANG

Lin WANG

Instructor: Zhi YAN

Summary

Summary	2
I. Introduction.....	3
A. What is a clustering algorithm?	3
B. Description of the project.....	3
1. Conditions connues	3
2. Objective	3
3. Approaches.....	3
II. Main body.....	4
A. K-means	8
1. Introduction of K-means.....	8
2. Application of the K-means algorithm.....	9
B. Meanshift	13
1. Introduction of Meanshift	13
2. Application of the Meanshift algorithm	13
C. DBSCAN.....	16
1. Introduction of DBSCAN (Density-Based Spatial Clustering of Applications with Noise)16	
2. Application of DBSCAN algorithm	18
III. Comparison of the three methods	21
A. How to measure the merits of clustering algorithms	21
B. Comparison of results.....	21
IV. Conclusion.....	23
V. Reference.....	24

I. Introduction

1.1. What is a clustering algorithm?

Clustering is the partitioning of a dataset into different classes or clusters according to a particular criterion (such as the distance criterion), such that the Data objects are as similar as possible, and at the same time as different as possible for data objects that are not in the same cluster. That is, after clustering, data of the same class are brought together as much as possible, and different data are separated as much as possible.

1.2. Description of the project

1.2.1. Conditions connues

Introduction

An important challenge that needs to be addressed to move towards a fully autonomous AGV is how to enable the AGV to correctly reach the designated location to perform corresponding tasks, including charging, loading and unloading cargo, receiving maintenance, and more. To do so, today's popular AGVs are equipped with various sensors to perceive the surrounding environment to identify the correct station. In this project, we hope to use the data obtained with 3D lidar to help AGV achieve automatic station alignment.

Available facilities

Several sets of points showing different environmental appearances (shapes). Each set of points corresponds to a text document (containing a set of coordinates in three-dimensional space) to describe a shape.

1.2.2. Objective

In order to better understand the appearance of the environment, we first use a clustering algorithm to segment the point cloud in order to better identify which points belong to which object. In this task, we need to compare different clustering algorithms in the field and select the one with the best performance.

Analyze the unified database with different algorithms and compare advantages and disadvantages

1.2.3. Approaches

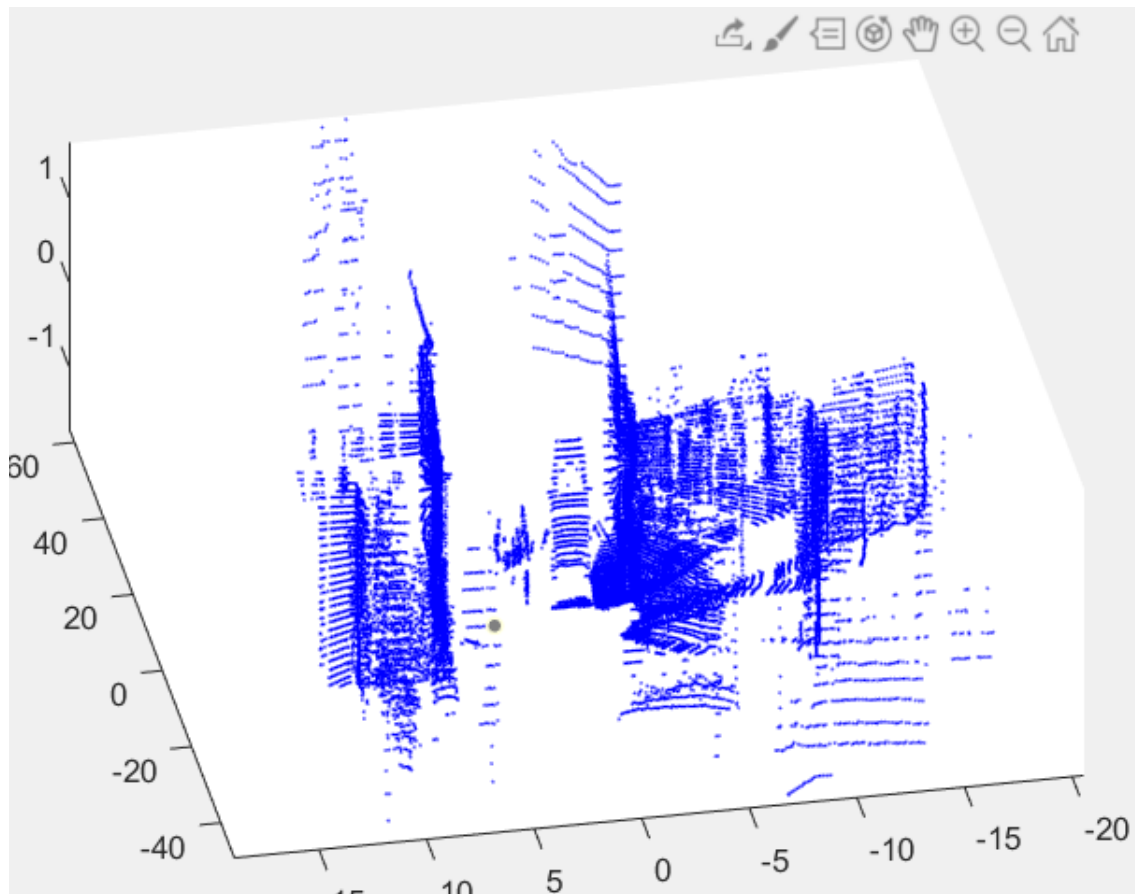
We apply three different clustering algorithms to cluster the databases. (**Kmeans**, **Meanshift**, **DBSCAN**)

2. Main body

A. Data Processing

1. 3D point visualization

We first select sample No. 3 as the analysis object, we get 113110 3D space points. When visualizing points in 3D space, I found many points on the ground. To see the result clearly, I delete all the data which blows to the earth. Then I visualize all the points as shown below.



2. Determine the position of the target object

In order to compare the pros and cons of various algorithms, we must first determine a standard, In the data provided by the teacher, we obtained the relevant parameters of the target object.

Car 0.00 0 1.55 614.24 181.78 727.31 284.77 1.57 1.73 4.15 1.00 1.75 13.22 1.62

DontCare -1 -1 -10 5.00 229.89 214.12 367.61 -1 -1 -1 -1000 -1000 -1000 -10

DontCare -1 -1 -10 522.25 202.35 547.77 219.71 -1 -1 -1 -1000 -1000 -1000 -10

Ignoring the latter two sets of data, we need to determine the meaning of the first set of data and find the actual location of the target.

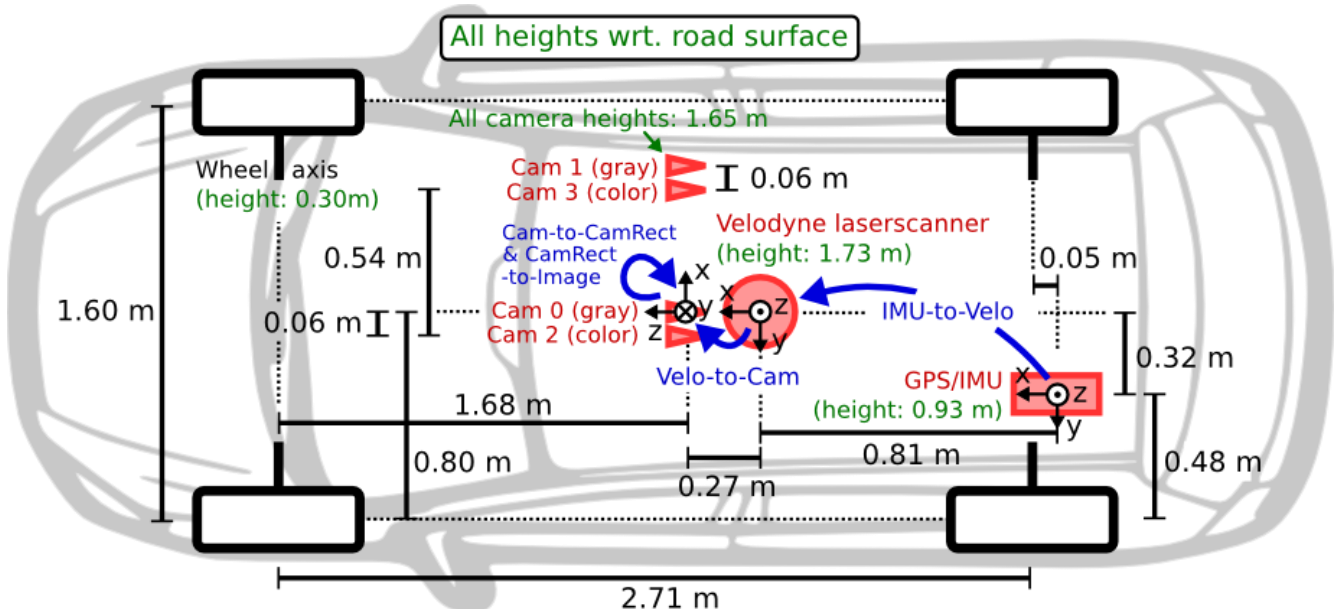
With the help of teacher YanZhi, we got a lot of information.

#Values	Name	Description
1	type	Describes the type of object: 'Car', 'Van', 'Truck', 'Pedestrian', 'Person_sitting', 'Cyclist', 'Tram', 'Misc' or 'DontCare'
1	truncated	Float from 0 (non-truncated) to 1 (truncated), where truncated refers to the object leaving image boundaries
1	occluded	Integer (0,1,2,3) indicating occlusion state: 0 = fully visible, 1 = partly occluded 2 = largely occluded, 3 = unknown
1	alpha	Observation angle of object, ranging $[-\pi..pi]$
4	bbox	2D bounding box of object in the image (0-based index): contains left, top, right, bottom pixel coordinates
3	dimensions	3D object dimensions: height, width, length (in meters)
3	location	3D object location x,y,z in camera coordinates (in meters)
1	rotation_y	Rotation ry around Y-axis in camera coordinates $[-\pi..pi]$
1	score	Only for results: Float, indicating confidence in detection, needed for p/r curves, higher is better.

There is a total of 16 data in each row. The meaning of each data is as shown in the figure. We understand the meaning of all parameters of the TXT file according to this figure.

The data we need to use is the second half, which represents the three-dimensional length of the target and the three-dimensional center point coordinates of the target in the camera coordinate system.

Since the data of the three-dimensional points we obtained are all in the laser coordinate system, we need to convert the coordinates of the target object in the camera coordinate system to the laser coordinate system.



As shown in the figure, we need to perform matrix transformation on the coordinates of the target according to the required parameters.

According to the data, we obtained the parameters of the rotation and translation matrix.

$R =$ <table style="margin-left: 40px;"> <tr><td>0.0075</td><td>0.0148</td><td>0.9999</td></tr> <tr><td>-1.0000</td><td>0.0007</td><td>0.0075</td></tr> <tr><td>-0.0006</td><td>-0.9999</td><td>0.0148</td></tr> </table>	0.0075	0.0148	0.9999	-1.0000	0.0007	0.0075	-0.0006	-0.9999	0.0148	$T =$ <table style="margin-left: 40px;"> <tr><td>-0.0041</td></tr> <tr><td>-0.0763</td></tr> <tr><td>-0.2718</td></tr> </table>	-0.0041	-0.0763	-0.2718
0.0075	0.0148	0.9999											
-1.0000	0.0007	0.0075											
-0.0006	-0.9999	0.0148											
-0.0041													
-0.0763													
-0.2718													

We transform the coordinate system according to the three-dimensional point of the target, according to the formula $B=R*A+T$.

Car 1.57 1.73 4.15 1.00 1.75 13.22	$A =$ <table style="margin-left: 40px;"> <tr><td>1.0000</td></tr> <tr><td>1.7500</td></tr> <tr><td>13.2200</td></tr> </table>	1.0000	1.7500	13.2200	$B =$ <table style="margin-left: 40px;"> <tr><td>13.2475</td></tr> <tr><td>-0.9755</td></tr> <tr><td>-1.8264</td></tr> </table>	13.2475	-0.9755	-1.8264
1.0000								
1.7500								
13.2200								
13.2475								
-0.9755								
-1.8264								

Then we use the plot3 function to draw the cuboid.

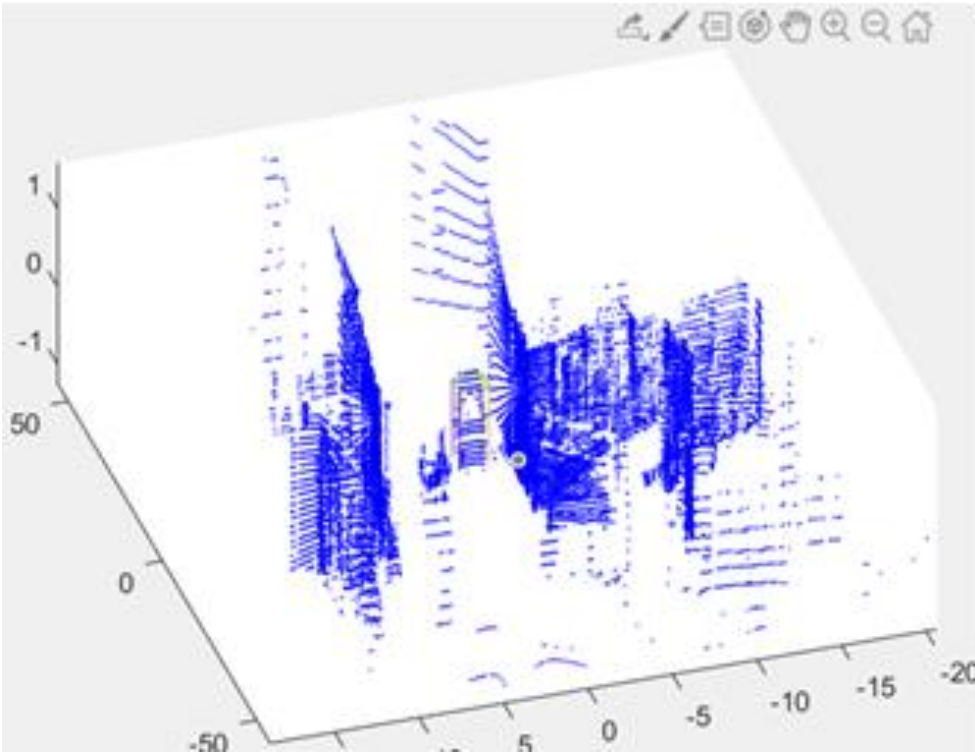
```
function boxplot3(x0, y0, z0, Lx, Ly, Lz)
% (x0, y0, z0) is the position of the first vertex;
% (Lx, Ly, Lz) is the length, width and height of the cuboid.
x=[x0 x0 x0 x0 x0+Lx x0+Lx x0+Lx x0+Lx];
y=[y0 y0 y0+Ly y0+Ly y0 y0 y0+Ly y0+Ly];
z=[z0 z0+Lz z0+Lz z0 z0 z0+Lz z0+Lz z0];
index=zeros(6, 5);
index(1, :)= [1 2 3 4 1];
index(2, :)= [5 6 7 8 5];
index(3, :)= [1 2 6 5 1];
index(4, :)= [4 3 7 8 4];
index(5, :)= [2 6 7 3 2];
index(6, :)= [1 5 8 4 1];
for k=1:6
plot3(x(index(k, :)), y(index(k, :)), z(index(k, :)))
hold on
end
```

```
boxplot3(11.1725, -1.8405, -1.8264, 4.15, 1.73, 1.57)
```

This function draws the cuboid with the point in the upper left corner as the starting point. We need to transfer the coordinates of the center point of the target to the coordinates of the upper left corner.

```
boxplot3(11.1725, -1.8405, -2.6114, 4.15, 1.73, 1.57)
```

We subtracted the coordinates of the center point by half the length, width and height of the cuboid. Then we draw the rectangle in 3D space.



As shown in the figure, we can see that the three-dimensional point representing the car has been framed by the cuboid.

Next we use algorithms to perform cluster analysis

3. Calculation of coincidence

In the evaluation system of target detection, there is a parameter called IoU, which is simply the overlap rate between the target window generated by the model and the original labeled window. Specifically, we can simply understand that: the intersection of the detection result (DetectionResult) and Ground Truth is compared to their union, which is the accuracy of detection IoU:

$$IoU = \frac{DetectionResult \cap GroundTruth}{DetectionResult \cup GroundTruth}$$

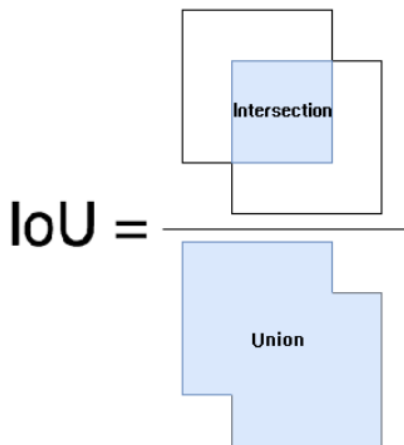


Diagram explaining IoU (from Wikipedia)

B. Data clustering

1 K-means

a) Introduction of K-means

Description of K-means

The k-means algorithm takes **k** as a parameter and divides **n** objects into **k** clusters so that the clusters have high intra-cluster similarity and low inter-cluster similarity.

The k-means algorithm is processed as follows:

First, **k** objects are randomly selected, each initially representing the mean or center of a cluster. For each remaining object, assign it to the nearest cluster based on its distance from the center of each cluster; then recalculate the average value of each cluster. This process is repeated until the criterion function converges.

In general, the squared error criterion is used, which is defined as follows:

$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2$$

E is the sum of the squared errors of all objects in the database.

p is a point in space.

m_i is the average of the clusters **C_i**.

The objective function makes the generated clusters as compact and independent as possible, using the Euclidean distance measure, although other distance measures can also be used.

a. Algorithmic progression

Input: the number **k** of databases and clusters containing **n** objects.

Output: **k** clusters to minimize the squared error criterion.

1. A random selection of **k** objects, each of which initially represents the centre of a cluster.
2. For each remaining object, assign it to the nearest cluster according to its distance from the centre of each cluster.
3. Recalculate the average value of each cluster and update it to the new cluster centre.
4. Keep repeating 2 and 3 until the criterion function converges.

b. Advantages and disadvantages

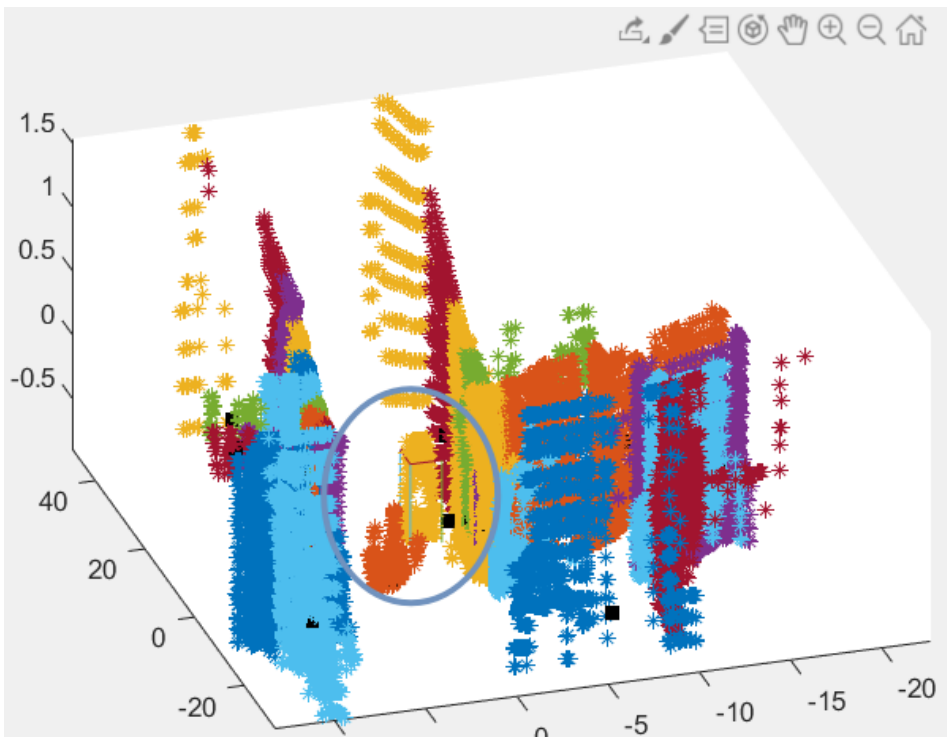
Advantages: simple and efficient for large datasets as well, with low time and space complexity.

Disadvantages: the most important is that the result is easily localized when the data set is large; need to set the K value in advance, sensitive to the first K point selection

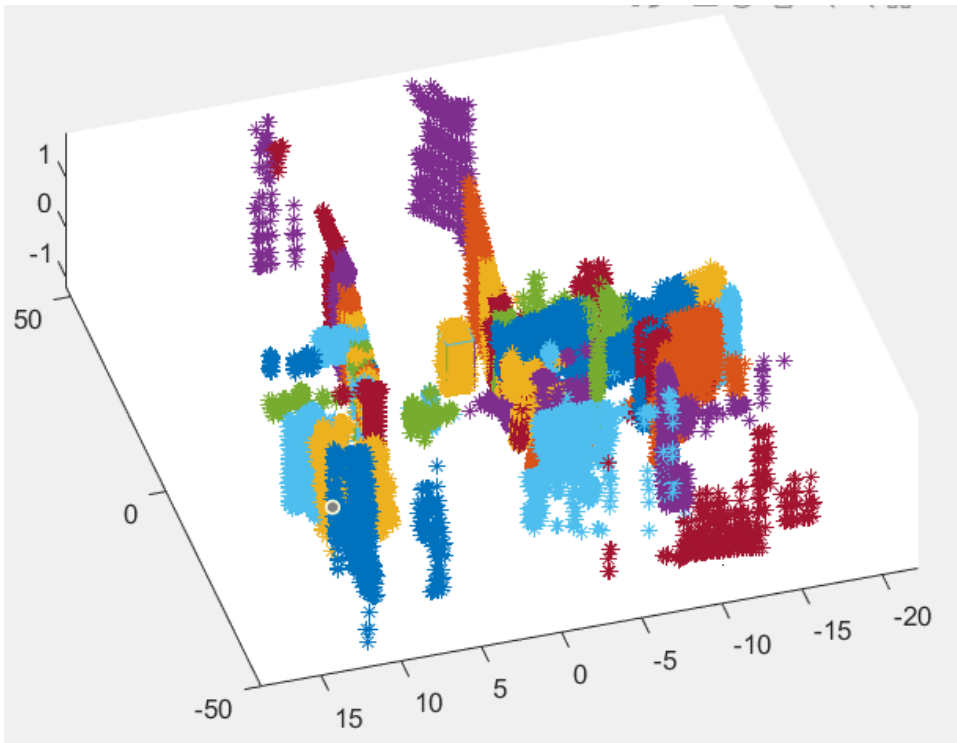
b) Application of the K-means algorithm

We first adopt the K-Means algorithm for cluster analysis, we want to get the most accuracy cluster by adding the value of K. For the selection of the initial point, we choose random selection.

First, we chose K=40, we can see the center of the cluster is near the car. But there are still a lot of points that do not belong to the car category. There is no doubt that the clustering we get when K=40 is very inaccurate and has no value to analyze.



Then, we choose K=80 for the second set of analysis. We can see that the points of the car classification have been greatly reduced, and we have obtained a better clustering effect.



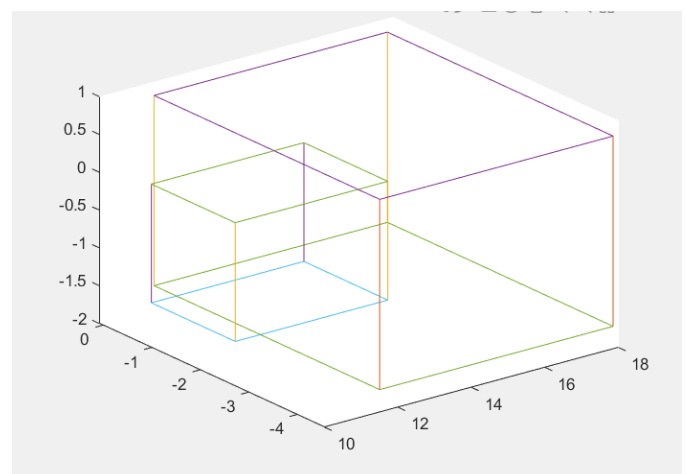
We analyze this cluster separately, calculate its length, width and height, and then compare it with the actual target position to get the coincident volume.

xReality	yReality	zReality					
15.32	-0.1105	-0.2564					
11.17	-1.841	-1.826					
				Length	Width	Height	Volume
			Clustering	6.35	4.652	2.515	74.2936
xClustering	yClustering	zClustering	Reality	4.15	1.7305	1.5696	11.2722
17.844	0.078	0.817	Superposit	4.15	1.7305	1.4416	10.35296
11.494	-4.574	-1.698					
			Reliability	14%			
xSuperposition	ySuperposition	zSuperposition					
15.32	-0.1105	-0.2564					
11.17	-1.841	-1.698					

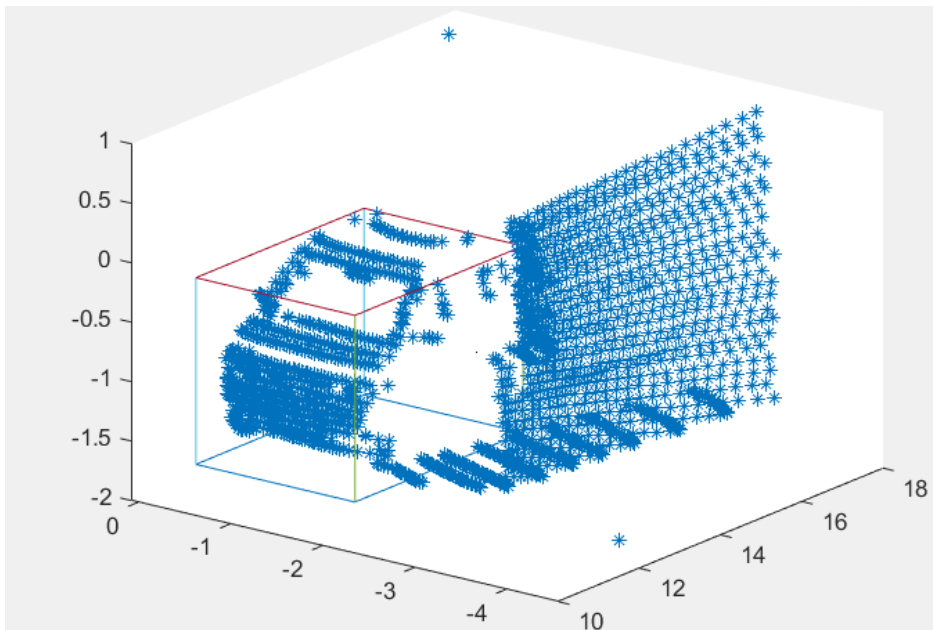
We calculate the actual volume of the target object, and then calculate the clustered volume and coincident volume. Then we use the IoU method to calculate the accuracy of this K.

The accuracy is equal to coincident volume / (actual volume + cluster volume - coincident volume).

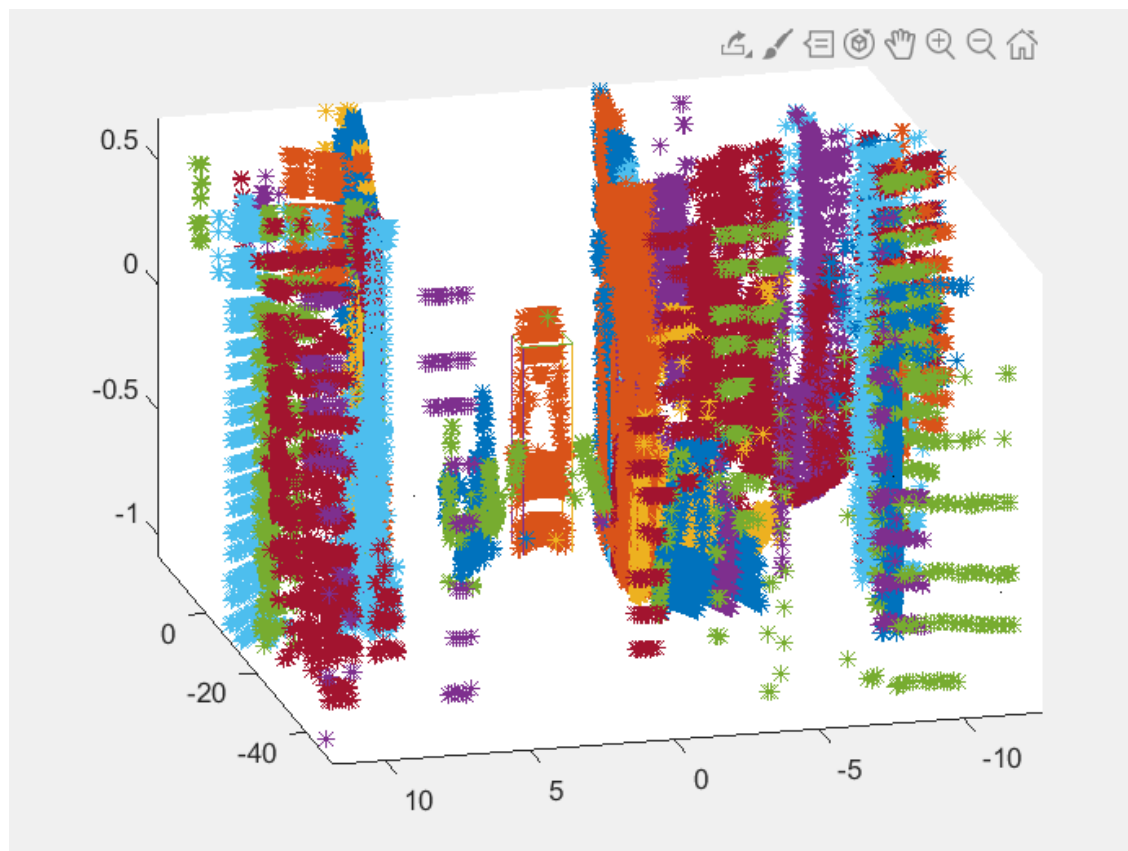
$$\text{Reliability} = 10.32 / (74.29 + 11.27 - 10.32) = 14\%$$



As shown in the following figure, the spatial points of the clustering when $K=80$, some points on the right side of the car are classified into this category. In order to improve accuracy, we increased the value of K .



When $K=80$, the data is lower than we expect, so we try to increase the value of K . After progressively increasing the value of K , we obtain that the Reliability is acceptable when $K = 150$. The following figure is the clustering result of $K=150$.

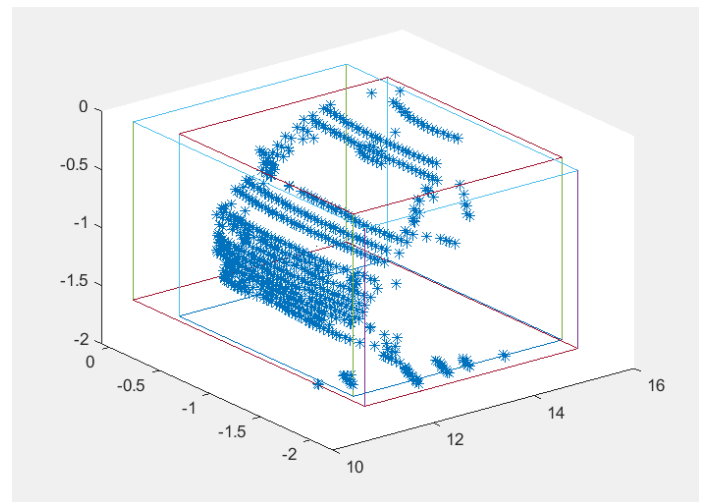
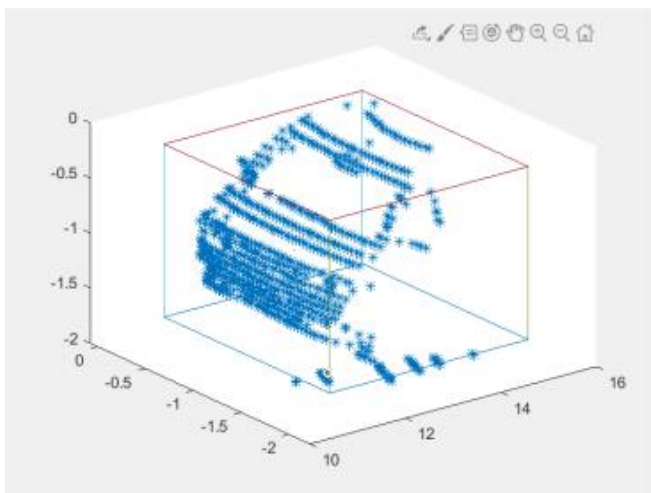


xReality	yReality	zReality						
15.32	-0.1105	-0.2564						
11.17	-1.841	-1.826						
xClustering	yClustering	zClustering		Length	Width	Height	Volume	
14.881	0.078	-0.166	Clustering	4.247	2.297	1.532	14.94521	
10.634	-2.219	-1.698	Reality	4.15	1.7305	1.5696	11.2722	
			Superposit	3.711	1.7305	1.4316	9.193571	
			Reliability	54%				
xSuperposition	ySuperposition	zSuperposition						
14.881	-0.1105	-0.2564						
11.17	-1.841	-1.688						

We use the same method to calculate the coincidence degree, and we get a credibility of 54% this time, we think this is a more accurate result.

Reliability=54%

As shown in the figure, the point of the target has been largely identified at $K = 150$. we can see that there are not many spatial points in the rectangular body of the cluster that do not belong to the car. Using the K-means algorithm can get a better analysis result.



If we continue to increase the value of K , we can further obtain more accurate results, but when the credibility is greater than 50, we think this is an acceptable result.

B. Meanshift

1. Introduction of Meanshift

1) Description of Meanshift

MeanShift is primarily an algorithm for implementing mass-center drift using Gaussian kernel functions.

The algorithm does not need to define the number of clusters, but only the radius of the center of mass circle, after which by calculating the mean of the vector distance from the center of mass to all points in the circle, if the mean distance of other points in the circle as the center of mass are less than the center of mass, then the circle will not continue to move.

In the Mean Shift algorithm, the probability density is actually used to find the local optimal solution to the probability density.

2) Advantages and disadvantages of Mean Shift algorithm

advantage:

No need to set the number of clusters.

Can handle clusters of any shape.

The algorithm only needs to set the bandwidth parameter, which affects the kernel density estimation of the data set.

The algorithm result is stable, no sample initialization like K-means is needed.

Disadvantages:

The clustering result depends on the bandwidth setting. The bandwidth setting is too small, the convergence is too slow, and the number of clusters is too large; the bandwidth setting is too large, and some clusters may be lost.

For larger feature spaces, the amount of calculation is very large.

2. Application of the Meanshift algorithm

We use the software **anaconda navigator**, where the **Jupyter Notebook** is a web application that facilitates the use of the **Jupyter Notebook module**. Create and share program documentation, support real-time code, mathematical equations, visualizations, and more. Uses include: data cleaning and conversion, numerical simulation, statistical modeling, machine learning, and more.

In our case, we open the folder where the code and the database are stored, and then we can perform the calculations from the web. To implement **DBSCAN** and **meanshift** clustering, we used `sklearn.cluster` Database. `scikit-learn` is a machine learning tool based on the Python language. All you need to do is import the database of clustering types and enter the required parameters to get the clustering results you need. This greatly simplifies our workload.

➤ **core code**

clustering = MeanShift(bandwidth=4).fit(X)

(The **bandwidth** can be interpreted as the radius of the configured center of mass circle.)

```
In [*]: import pandas as pd
import csv
from sklearn.cluster import MeanShift
import numpy as np
import copy
from sklearn.cluster import DBSCAN

temp=np.loadtxt("aboveEarth.csv", dtype=np.int, delimiter=',')
temp = np.delete(temp, 0, axis=1)
temp = np.delete(temp, -1, axis=1)
X = temp
clustering = MeanShift(bandwidth=3).fit(X)
n = list(clustering.labels_)
```

```
In [*]: data = pd.read_csv("3aboveEarth.csv")
data["n"] = n
data.to_csv("newMean.csv")
```

➤ **Results**

When bandwidth=4 clustering to 42

When bandwidth=3 clustering to 71

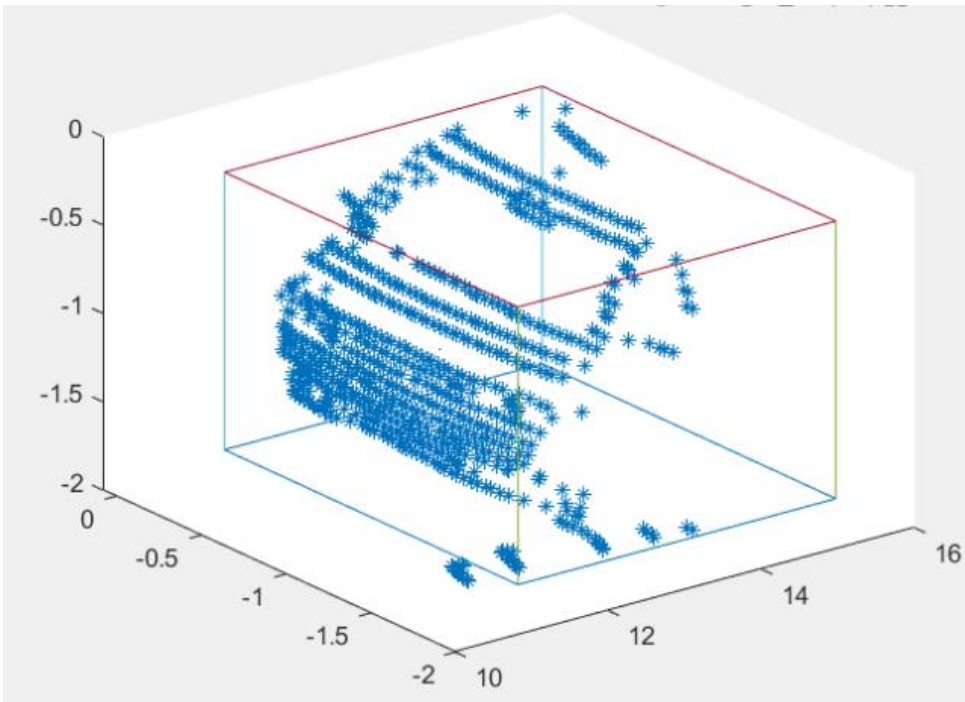
When bandwidth=2 clustering to 126

We use python to perform Meanshift cluster analysis on the data, We take the boundary as 2, 3, and 4 respectively for cluster analysis, When the Bandwidth is 3 and 4, there are too few categories, and the target cluster contains a large number of other points, which is not suitable for analysis.

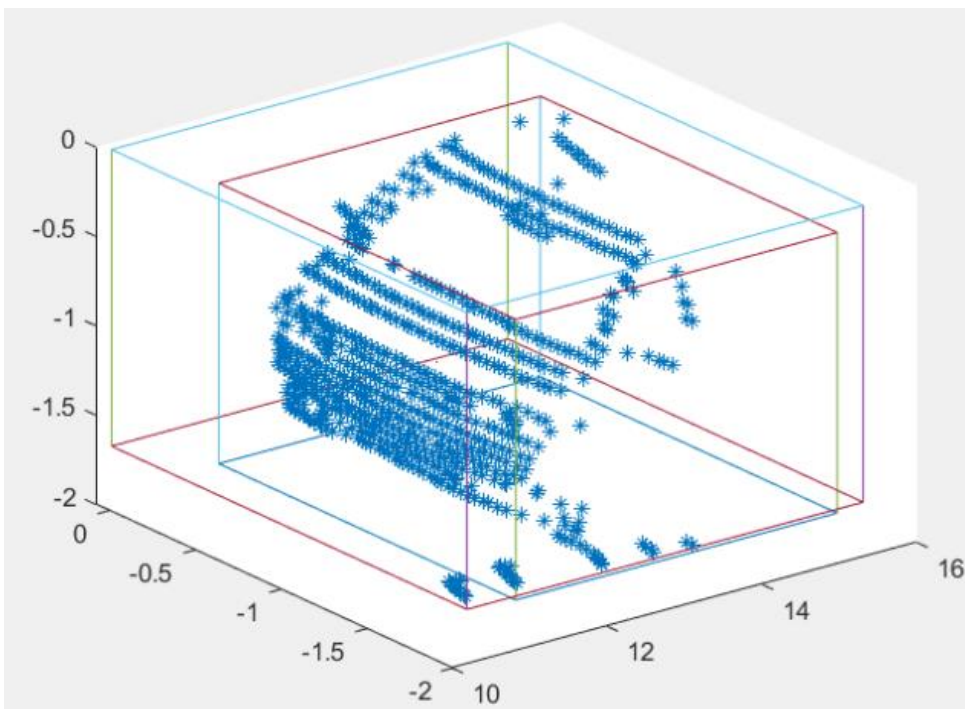
When bandwidth=4, we got 126 clusters, at this time, we can conduct further analysis. We extract the clusters containing the target objects and put them into Matlab for further analysis. At the same time, we also use the IoU method for coincidence test.

xReality	yReality	zReality					
15.32	-0.1105	-0.2564					
11.17	-1.841	-1.826					
				Length	Width	Height	Volume
			Clustering	4.677	2.071	1.532	14.83905
xClustering	yClustering	zClustering	Reality	4.15	1.7305	1.5696	11.2722
14.881	0.078	-0.166	Superposit	3.711	1.7305	1.4416	9.25779
10.204	-1.993	-1.698					
			Reliability	55%			
xSuperposition	ySuperposition	zSuperposition					
14.881	-0.1105	-0.2564					
11.17	-1.841	-1.698					

We got a credibility of 55%, we think it is a value that can identify objects, so Meanshift algorithm very quickly obtained an acceptable result.



We put the obtained clustering results into Matlab for visualization. As shown in the figure, we got a pretty good result.



C. DBSCAN

1. Introduction of DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

a. Description of DBSCAN

DBSCAN is a density-based clustering algorithm, which generally assumes that categories can be distributed through the sample distribution of the closeness decision. Samples of the same category, their close proximity to each other. That is, there must be samples of the same category within a short distance of any sample in that category.

By classifying closely related samples into one category, this gives us a clustering category. By classifying all groups of closely related samples into different categories, we get the final result of all clustered categories.

In intuitive effect, the DBSCAN algorithm finds all the dense regions of the sample points and treats these dense regions as clusters one by one.

The DBSCAN algorithm has the following features.

- Density-based, robust to noise points away from the density core.
- No need to know the number of clusters.
- Clusters of arbitrary shape can be found.

b. Parameters

To facilitate the calculation, we only used the following two parameters

1) eps: DBSCAN algorithm parameters, i.e., the distance threshold for our ϵ -neighborhood, and the sample distance over ϵ . The point is not in the ϵ -neighborhood. The default value is 0.5. It is usually necessary to select a suitable threshold value by choosing among multiple values. eps is too large, more points will fall in the ϵ -neighborhood of the core object, our number of categories may be reduced, and samples that should not be in a category will also be classified as one category. Conversely, the number of categories may increase, and samples that were originally in one category are divided apart.

(2) min_samples: the DBSCAN algorithm parameter that the sample point needs to become a core object ϵ - The sample count threshold for the neighboring domains. The default value is 5. It is usually necessary to select an appropriate threshold value by selecting a value within a multiple set of values. This is usually done in conjunction with eps. In the case of certain eps, if min_samples is too large, then the core objects will be too few, and the inner part of the cluster would have been Samples in one category may be marked as noise points, and the number of categories will become high. On the other hand, if min_samples are too small, a large number of core objects will be created, which may result in a low number of categories.

c. Algorithmic progression

The DBSCAN algorithm is divided into two steps.

1, find the core points to form temporary clusters.

Scan all the sample points, and if the number of points within the radius R of a sample point is \geq MinPoints, they are included in the list of core points with direct density to form the corresponding temporary clusters.

2, merge the temporary clusters to obtain the cluster clusters.

For each temporary cluster, check whether the point in it is a core point, and if so, merge the temporary cluster corresponding to that point with the current temporary cluster to get the new temporary cluster.

Repeat this operation until every point in the current temporary clustering cluster is either not in the core point list or its density direct is already in that temporary cluster, which is upgraded to a cluster. Continue the same merge operation for the remaining temporary cluster clusters until all temporary clusters are processed.

d. Advantages and disadvantages

The main advantages of DBSCAN :

- 1) It is possible to cluster dense datasets of arbitrary shape, in contrast, clustering algorithms such as K-Means are generally only applicable to convex datasets.
- 2) Anomalies can be found while clustering, which is insensitive to anomalies in the dataset.
- 3) The clustering results are unbiased; relatively, the initial values of clustering algorithms such as K-Means have a great impact on the clustering results.

The main disadvantages of DBSCAN :

- 1) If the density of the sample set is not uniform and the cluster spacing difference is very different, the quality of clustering is poor, then clustering with DBSCAN is generally not suitable.
- 2) If the sample set is large, the clustering convergence time is long, which can be improved by limiting the size of the KD tree or sphere tree established when the nearest neighbor is searched.
- 3) Compared with the traditional K-Means clustering algorithm, it is a little more complicated, which requires joint tuning of distance threshold ϵ and neighborhood sample number threshold MinPts, and different parameter combinations have a great impact on the final clustering effect.

2. Application of DBSCAN algorithm

a. Using Anaconda Jupyter

To implement the algorithm, we programmed with **Anaconda Jupyter**.

What needs to be constantly adjusted are the two parameters, **eps** and **min_samples**, and the combination of these two values has a significant impact on the final clustering effect.

Code

```
In [2]: import pandas as pd
import csv
from sklearn.cluster import MeanShift
import numpy as np
import copy
from sklearn.cluster import DBSCAN

temp=np.loadtxt("aboveEarth.csv", dtype=np.int, delimiter=',')
temp = np.delete(temp, 0, axis=1)
temp = np.delete(temp, -1, axis=1)
X = temp
clustering = DBSCAN(eps=0.5, min_samples=250).fit(X)
n = list(clustering.labels_)
```

Then, we insert an excel sheet to cluster the totals

Assigning values to two parameters

eps=0.5, min_samples=300 Clustering → 74 classes

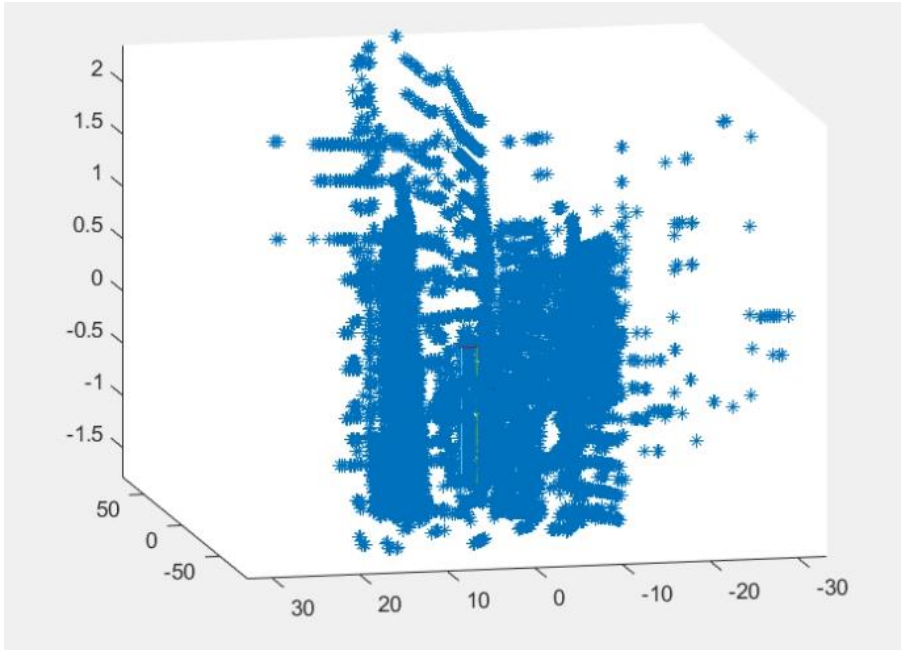
eps=0.5, min_samples=250 Clustering → 87 classes

eps=0.5, min_samples=200 Clustering → 104 classes

By comparison, it was found that changing the **eps** value had **no effect** on the clustering of our database, so we only changed the **min_samples**, **the smaller the min_samples, the more clustering**. eps and min_samples have default parameters of 0.5 and 5, but it was found through experimentation that this parameter yields too much clustering and splits the target object into parts.

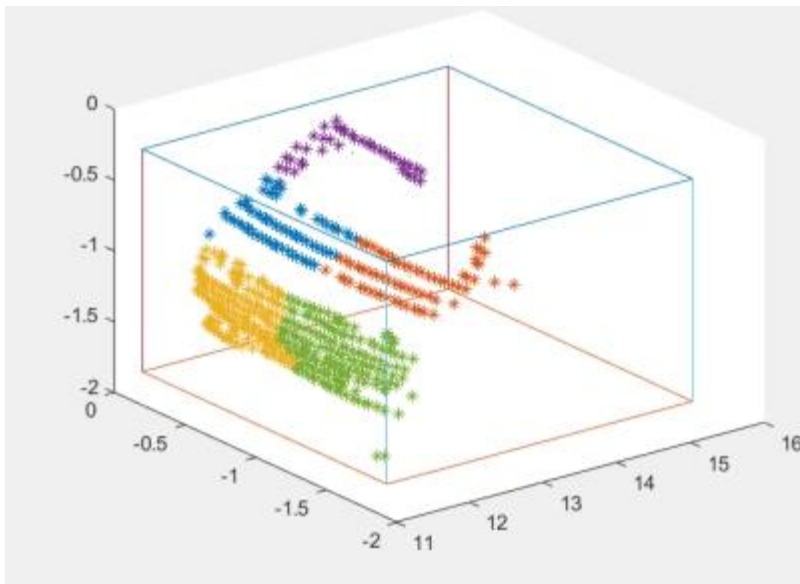
b. Using matlab

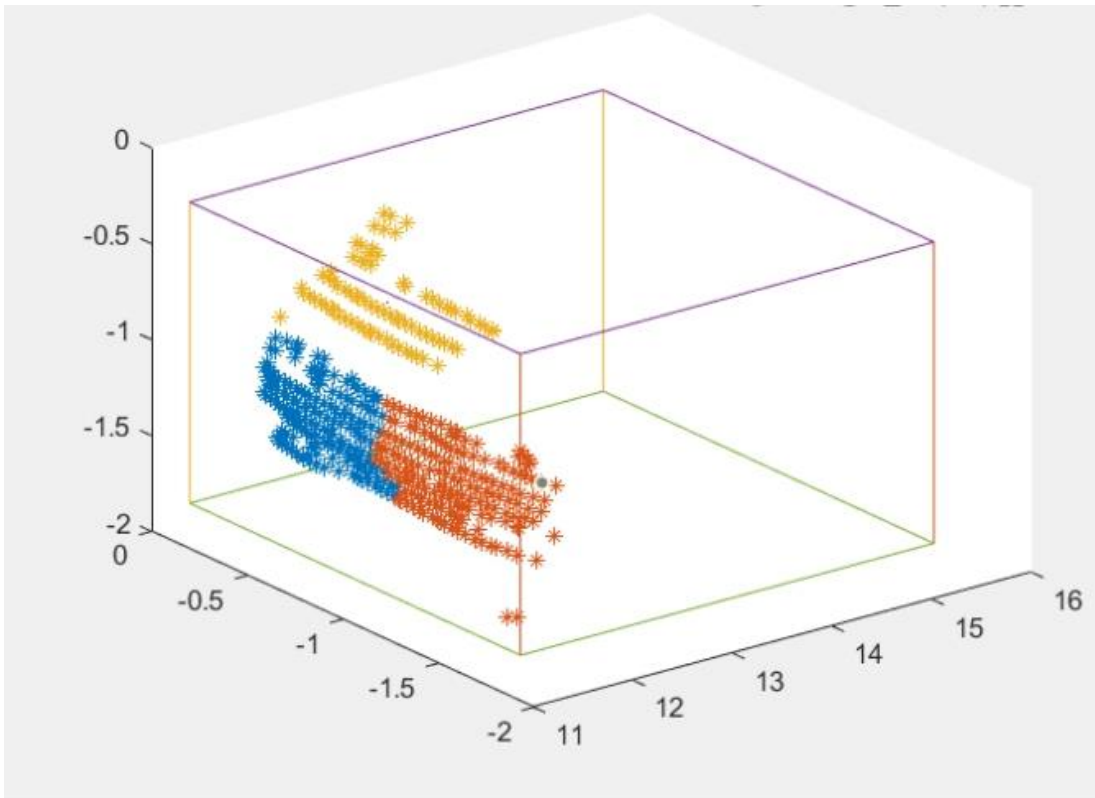
- Assigning values $\text{eps}=0.5$, $\text{min_samples}=200$ Clustering → 105 classes



The clustering effect is not very effective and does not identify the needed objects.

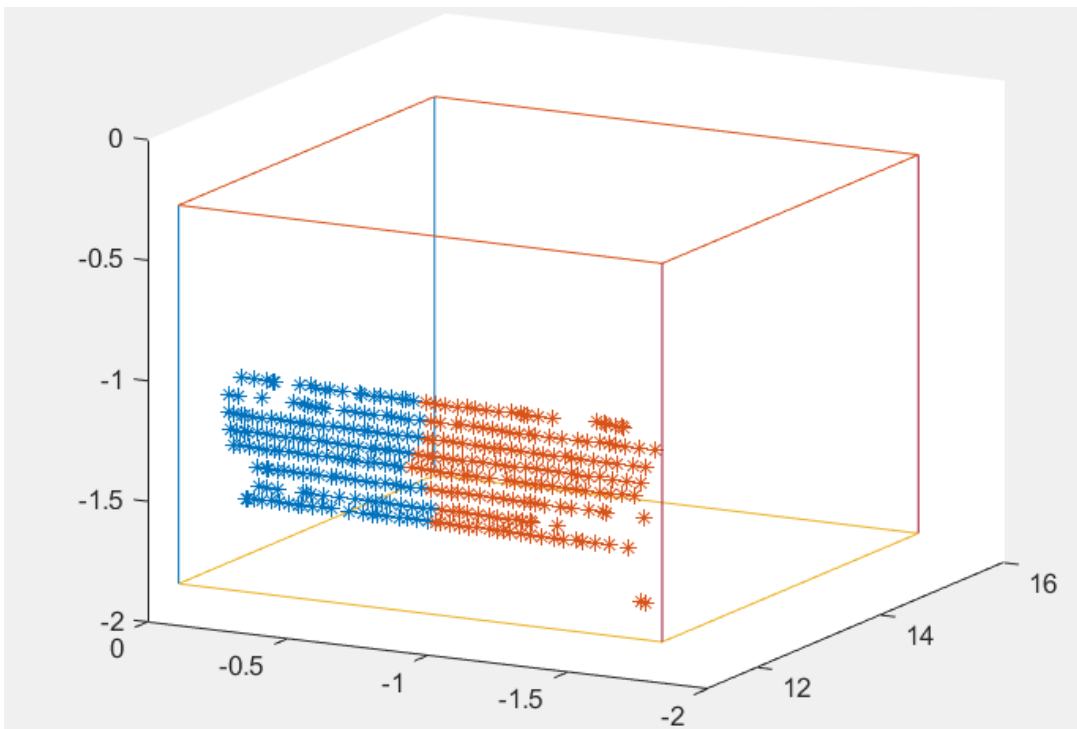
- Assigning values $\text{eps}=0.5$ $\text{min-sample}=50$
Five clusters construct most of the car's feature points, and many more points are grouped into other classes. The results are not very effective. No accurate clustering.





Assigning values $Eps=0.5$ $min_sample=80$

Three clusters construct most of the car's feature points, and many more points are grouped into other classes. The results are not very effective. No accurate clustering.



$Min_sample=100$ works ineffectively.

III. Comparison of the three methods

A. How to measure the merits of clustering algorithms

Different clustering algorithms have different advantages and disadvantages and different applicability conditions. Broadly in terms of the properties of the heel data (whether or not it is serially input, dimensionality), the presuppositions of the algorithm model, and the processing power of the model. They are as follows.

- 1. the processing power of the algorithm:** the ability to handle large data sets (i.e., the complexity of the algorithm); the ability to handle data noise; the ability to handle arbitrary shapes, including nested data with gaps.
- 2. whether the algorithm requires pre-conditions:** whether the number of clusters needs to be known in advance, whether the user needs to give domain knowledge.
- 3. data input attributes of the algorithm:** whether the results processed by the algorithm are related to the order of data input, that is, whether the algorithm is independent of the order of data input; the ability of the algorithm to process data with many attributes, that is, whether it is sensitive to the number of data dimensions and whether it has requirements for the type of data.

B. Comparison of results

- The difficulty with the K-means method is the value of K. Different K's affect the reliability of the detection results.
- Meanshift clusters quite well simply by changing the width of the selection. Compared to K-means, it is easier to identify objects with similar accuracy as K-means.
- The DBSCAN compares poorly with the other two algorithms and is almost impossible to obtain satisfactory results.

C. Analysis of results

1) K-means: it clusters with k points in space as the center, classify the objects closest to them, and iteratively update the value of each cluster center successively until the best clustering result is obtained.

When the K-MEANS algorithm is used to process a large number of three-dimensional points in space, it is possible to cluster similar points well, and obtain more reliable results for object detection.

For K-means, it is well interpretable (e.g., when it is necessary to create a taxonomy), and there is also some research showing that these algorithms produce high quality clustering, which is also applied in the merging phase after first taking K-means that are larger.

2) Meanshift: It does not need to set the number of clusters; it can handle clusters of any shape.

The algorithm only needs to set the bandwidth parameter, which affects the kernel density estimation of the data set.

The Mean Shift algorithm is most commonly used for target tracking. It calculates the probability density distribution of the similarity between the candidate target and the target template, and then uses the direction of the probability density gradient decline to obtain the best path for matching search, accelerating the positioning and reduction of moving targets. The search time, therefore, has high application value in the real-time tracking of targets.

When it is at a point in the clustering space, due to the different density of each object, the Meanshift algorithm can be used to obtain good clustering results very quickly.

Density-based methods: k-means cannot solve the clustering of irregular shapes. So density-based methods were developed to systematically solve this problem.

For meanshift: the entire end result is controlled by one parameter—the kernel bandwidth value. Other clustering approaches, such as k-means, require a number of clusters to be specified as an input. This is acceptable for certain scenarios, but most of the time the number of clusters is not known.

3) DBSCAN: We cannot get a good result when using the DBSCAN algorithm, We found through analysis that the algorithm uses unified eps and Min-samples throughout the clustering process. This method will not have a great impact on the clustering result when the data distribution is relatively uniform; but if the data distribution is significantly uneven, it will cause the clustering effect to appear Large deviation. Lidar output is a typical form of unevenly distributed data. The data density closer to the origin of the lidar is higher, while the data density further away from the origin of the lidar is lower. This difference in density has nothing to do with the size of the obstacle but is caused by the lidar measurement mechanism. When the traditional DBSCAN algorithm is used to cluster lidar data, there will always be a distance clustering effect, that is, no matter what ϵ and Min-samples value is selected, some distance areas will miss or mis detect.

IV. Conclusion

In this project, we have a deeper understanding of the three clustering algorithms. We have mastered their principles and implementation methods, and their application in actual situations. They each have different advantages and disadvantages, suitable for different data analysis. We use the "trial and error method" to determine the K value, bandwidth value, eps and MInsample value, and we use this method to approach the correct result a little bit. We have a more intuitive understanding of the core parameters of different algorithms, and their impact on the algorithm is also shown to us through the results.

In our project, we need to analyze the points in space. We have mastered the Matlab tool to process and visualize the points. We have learned a lot of knowledge through a lot of literature search and reading. For example, the implementation of K-means algorithm in Matlab, such as the implementation of K-means algorithm in Matlab, how to use python to call the clustering algorithm. Through this project we have mastered a variety of analysis tools and analysis methods, which is a great gain for us.

At the same time, we also need to summarize the errors in this project. We spent a lot of time researching the Kitti data set. When converting the coordinate system, we did not find the correct method very quickly. Fortunately, with the help of experiments and teachers, we finally found the correct use of the parameters and the coordinate conversion between the camera coordinate system and the laser coordinate system.

We also need to reflect on the lack of awareness of the project at the beginning. We spent a lot of time to find Matlab code that implements various algorithms but ignored our goal to use various algorithms to analyze data. In the mid-term, we realized Until we can directly call the python database to perform cluster analysis on the data directly. In the future, we should focus more on the determination of goals.

In short, this project has given us a lot of gains. We learned a scientific method of analyzing problems: trial and error. We have mastered excellent analysis tools: Matlab and python. Through them, we can achieve data visualization and cluster analysis. We understand the principles, advantages and disadvantages of the three algorithms, and have a clearer understanding of the different algorithms for different occasions.

Thanks to teacher YAN Zhi for giving us guidance and help in this project, we have successfully achieved the expected goal.

V.Reference

<https://blog.csdn.net/zhoub1668/article/details/7881313>
<https://blog.csdn.net/abc200941410128/article/details/78541273>
<https://www.cnblogs.com/pinard/p/6217852.html>
<https://zhuanlan.zhihu.com/p/88747614>
<https://www.cnblogs.com/pinard/p/6208966.html>
<https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/>
<https://blog.csdn.net/zhouxianen1987/article/details/68945844>
<https://www.static.ixueshu.com/document/10a911bfed7419ce6d63e46ef0831adc318947a18e7f9386.html>
<https://www.cnblogs.com/hdu-2010/p/4621258.html>
http://www.ojournal.org/mv_html/j00001/2019-07/A190714000010_WEB.htm
<https://www.cnblogs.com/wangyarui/p/6526300.html>
<https://cloud.tencent.com/developer/article/1459530>
<https://blog.csdn.net/carson2005/article/details/7341051>
<https://blog.csdn.net/u010536377/article/details/50884416>
<https://gist.github.com/davidstutz/88a12b73813d0e054ece8ab1b53e58a9>
<https://github.com/godspeed1989/kitti-velodyne-viewer/blob/master/kitti.py>
https://github.com/CognitiveRobotics/omnimapper_matlab/blob/master/matlab/devkit/matlab/run_demoTracklets.m#L126
<https://blog.csdn.net/cuichuanchen3307/article/details/80596689>
https://link.springer.com/epdf/10.1007/s10514-019-09883-y?author_access_token=bVLnE4rWjkyUnk8WopA0Lfe4RwLQNchNByi7wbcMAY6zHIY15ykgJsK70R8O7eQrMr2yHIZQSiye3OktHw_9R1puJtMefwAs4tGo2L7ytrEzPSDTxHtSdjXNYkRozK46fQM7ZPLogSknycKxSolsA%3D%3D
https://blog.csdn.net/qq_31622015/article/details/90678908
https://blog.csdn.net/Eddy_zheng/article/details/52126641
<https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/>
<https://www.ijcsmc.com/docs/papers/July2013/V2I7201338.pdf>
<https://stackoverflow.com/questions/35952124/how-to-choose-initial-centroids-for-k-means-clustering>
<https://datascience.stackexchange.com/questions/5656/k-means-what-are-some-good-ways-to-choose-an-efficient-set-of-initial-centroids>
<https://github.com/gpolic/kmeans-excel>