International Journal of Future Generation Communication and Networking
Vol. 13, No. 2, 2020pp.577-584

A Hybrid Machine Learning Approach for Efficient Deployment of Virtual Screens based on Unity

M.Vijayasanthi[1], A V L N Sujith[2], Saritha Anchuri[3], P.Lakshmi[4], B.Rajini[5]

[1, 3]Associate Professor, [2,4, 5]Assistant Professor

[2]Gates Institute of Technology, Gooty, Ananthapuramu, Andhra Pradesh, India

[1, 3, 4, 5]S V Engineering College (SVEC), Tirupati, Andhra Pradesh, India

Abstract

In the current generation computing technology, the virtual screens of the unity are generated in a manual process. With the advent of machine learning the automation of computing paradigms are leveraged through the enforcement of deploying efficient machine learning algorithms. The main objective of this article is to properly engaging efficient machine learning algorithm for rapid deployment of unity virtual screens. Original k-means clustering algorithm chooses initial clustering centroids randomly, and it may lead the converge of clustering results to local optimal solution. According to the Unity terrain data distribution, the scope of the initial clustering centroids can be determined, so we choose k most remote data point as the initial clustering centroids. The experimental results show that the algorithm can guarantee the clustering quality and reduce the number of iterations effectively, and it can be used to generate Unity virtual scenesautomatically.

Keywords: *Virtualization, virtual screens, Automation, Remote computing, Artificial intelligence*

1. Introduction

Virtual scene is an abstraction of the real world. With the development of virtual technology,Virtualscenewaswidelyusedinmanyways,forexample,inIndustrialequipment maintenance, Farming simulation, Patients medical rehabilitation, Natural disaster assessment, Virtual museum [1]-[5]. Following the advanced computer hardware, the scale of virtual scene that could simulate will be increasing. There are variety virtual scene models, and it will be a cumbersome task when the position, direction and other relevant properties were required to be adjusted for each model object. Dealing with the model object in simple rules can lead to an excessive neat or duplicateresult.

According to the above issues, this paper proposes a virtual scene fast generation method based on Machine Learning. The key of virtual scene generated is to place the corresponding models in specific coordinate points. Firstly it need to segment the Unity terrain and acquire the coordinate points, then those points are managed with Machine Learning algorithms and make sure that those points are used to place the same model belong to the same class. At last, we can instantiate models for those points in the same class and generate virtual scenequickly.

2. Related Work

The methods of building virtual scene consist of general 3D modeling software (3DSMax, Unity3D), virtual reality development kit and 3D engine toolkit. Building a virtual scene is a cumbersome task no matter what method we take. The proposed method focus on the location information of models in virtual scene, the purpose of dealing coordinate points with k-means is to increase the speed of building virtual scene.Comparedwithotheralgorithms,k-meanscanobtainvariousknowledgewithout priorknowledge[6]-[8].K-meansisverysimpleandwidelyusedbecauseofitsperfect performance.

Fortherapidgenerationofvirtualscene,Senet.alobtainedsmall-scalecrowdparticle information by real pictures and establish Wang Tile sets. Wang Tile was used to cover the scene area, then arbitrary scale crowd would be built rapidly [9]. Shan et.al proposed an adaptive snowfall occlusion diagram model, which was based on the view- point. It could update the occlusion relation between

ISSN: 2233-7857 IJFGCN

terrains and models in real-time and reduce the amount of calculations greatly in large-scale snow scene. It was conductive to the rapid generation of snow scene[10].

K-means algorithm has many applications in all aspects, Prabira et.al proposed a novel approach to resolve the robustness and capacity of image and video, messages bits were clustered and grouped together using K-means clustering and then the clustered message was embedded inside the cover medium by using direct mapping [11]. Fan et.al proposed an improved K-means algorithm, they effectively detected outliers based gird density .In the paper [12], students were clustered by the improved K-means algorithm and school administrators could acquire the characteristics of students in different group.

Aiming at the clustering of coordinate points in the rapid generation of Unity virtual scene, this paper proposes a novel approach to deal with those points. Those points are clustered and grouped together using K-means clustering, and the selection of K-means initial clustering centers was limited by sample distribution. The clustering performance of K-means algorithm is guaranteed and the convergence of the algorithm is accelerated.

3. Terrain data processing based on K-means clustering algorithm

Clustering data sets are obtained by using Unity terrain abstraction, the distribution function of clustering centers is determined according to the distribution of sample data. The initial clustering of K-means algorithm is determined by clustering center function instead of random selection

3.1 Traditional K-means clusteringalgorithm

Clustering attempts to divide the data sets into several disjoint subsets, and each of them is called a cluster [13]-[14]. The basic idea of K-means algorithm is to adjust the clustering center by iteration, and finally all the samples are divided into K different subsets. Formally, it is assumed that the sample set D= $\{x_{\&},x_{(},\ldots,x_{**}\}$contains M samples, each of which X_{-} =($x_{-\&},x_{-(},\ldots,x_{-/}$)isann-dimensionalfeaturevector,Clusteringalgorithmdividessamplesetsdintokdisjointclusters$\{C_2|l$=1,2, …,$k\}$anddisjoint clusters satisfy those conditions:

$C_{9}:\cap C_{2}=\emptyset(l\neq l^{>})$ (1)

K-means algorithm $C_{9}:\cap C_{2}=\emptyset(l\neq l^{>})$ uses the distance as similarity evaluation criteria [15], if the distance of two samples was relative small, they are considered similar.

Clusteringperformancemeasureisalsocalledclusteringvalidityindex.Forclusteringresults, some kind of metric is needed to evaluate the clustering performance. Intuitively, we hope that samples of same clusters are as similar as possible, and samples of different clusters are as different as possible. According to clustering results C=$\{C_{\&}, C_{(}, \ldots, C_{A}\}$, some formulas are defined as follows:

$avg(C) = 2/B|C| * (|C| - 1)C\sum_{\&S\text{-}TFS|U|}L_{H}(x_{-}, y_{-})$ (2)

$D_{WX/}BU_{-}, U_{F}C = L_{H}BU_{-}, U_{F}C, \sum_{\&S\text{-}S|U|}x$ (3)

avg(C) is the average distance between samples incluster C, $D_{WX/}BC_{-},C_{F}C$ is the distance between cluster $C_{-}$and cluster C_{F}. Obviously, we hope that the value of DBI is verysmall

ISSN: 2233-7857 IJFGCN

3.2 Obtaining clustering datasets

Unity terrain fits in Euclidean coordinate system. Through the gird, we can divide Unity terrain evenly into N parts. Suppose that $T_i$is a complete terrain (n $\in R^{/}$), N sub terrains ($t_{-*F}(0 \le i<n)$, $(0 \le j<n)$) are obtained by dividing the terrain and the sub terrains($t_{-*-}=(x_{-},y_{-})$)are represented by the lower left corner coordinates. The whole terrain is abstracted into a series of coordinate points and those points are used to form a data set $D_{/}$. T_i&&$D_i$satisfies the followingconditions Table 1 shows the Dataset D_{yx} and Fig.1 represents abstracting process.

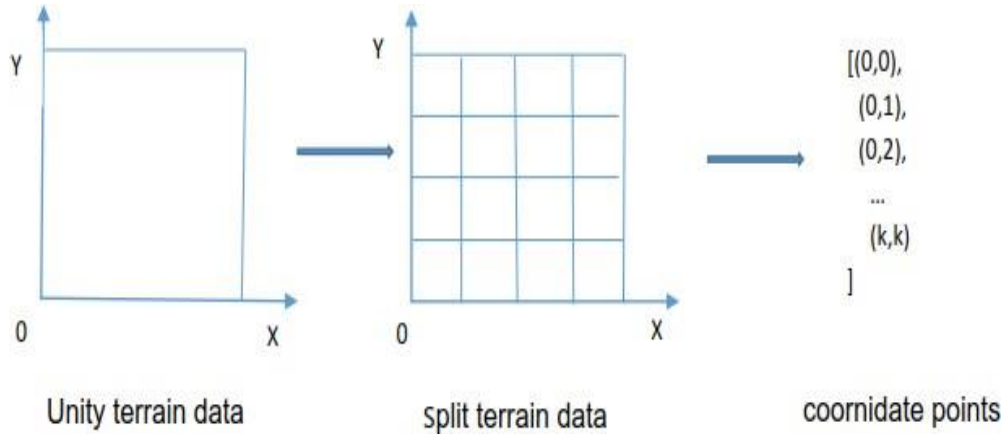

Fig.1 Terrain abstraction

Table1. DataSet

X/Y					
	(0,0)	(0,1)	……	……	(0,29)
	(1,0)	(1,1)	……	……	(1,29)
	……	……	……	……	……
	(29,0)	(29,1)	……	……	(29,29)

3.3 Optimize the initial clusteringcenters

The traditional K-means clustering randomly selects K initial clustering centers, which has a great influence on the clustering results. In view of the particularity of the uniform distribution of the terrain sample points, Clustering centers are obtained when the algorithm is convergent to satisfy certain distribution$\{(x_{-}, y_{-}) \mid y = f(x), i \in (0, k)\}$. If the selection of initial clustering centers is close to this distribution, the algorithm could reduce the number of iterations in the iteration process. The selection of the number of K-means clustering centers generally satisfies the conditions ($2 < k <\sqrt{m}$), m is the number of data points. We want to generate a scene rapidly in Unity, which consists 3 models.ThetraditionalK-meansclusteringalgorithmisusedtodealwithdataset(D_{*}), the value of k is 3, running n times to get n clustering centers. Taking the dataset D_{yx} as an example, the distribution of clustering centers shows infig.2:

ISSN: 2233-7857 IJFGCN

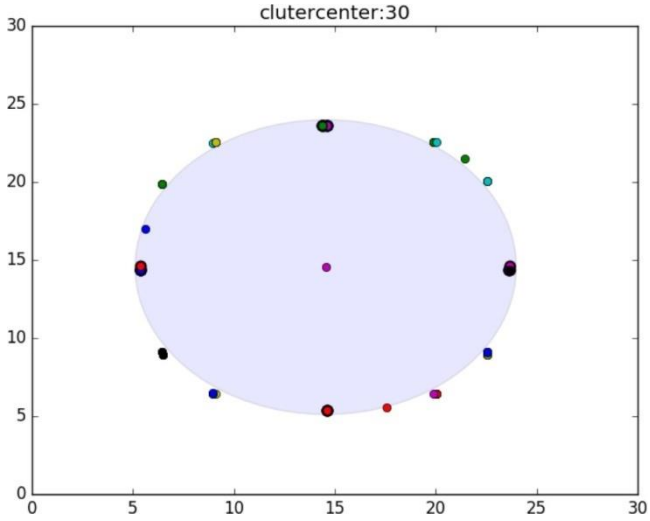

Fig2. Clustering Centroids distribution

30 times clustering analysis are given for data set D_{yx},and 90 clustering centers are obtained. Points which are around the circular boundary, are obtained by clustering algorithm, the center of the circle is located in the center of the whole data set. Through comparative analysis of several experiments, it is found that the distribution of clustering centers is similar to a circle, and this circle satisfies the followingconditions:

$$\{(x_{-},y_{-}) \mid (x-c-w)^{(}+(y-c-w)^{(}=(r-w^{\rangle})^{(}\}$$

For these formulas, c represents the center point of circle and w is the offset of the center point, r is the circle radius and $w^{\rangle}$is the offset of the circle radius, the values of c, w, r, $w^{\rangle}$depend on the size of data set $T_{/*/}$. Thus, the distribution function of initial clustering centers has been confirmed, and this function can be used to optimize the selection of initial clustering centers.

4. The K-means clustering algorithm based on Unity terraindata

Firstly, the selection of initial clustering centers is optimized according to the terrain data distribution. Each object in the data set is assigned to nearest cluster based on the distances between itself and each clustering center. After traversing all the objects, an iteration is completed and a new clustering center is calculated. If there is no change in the clustering centers before and after the iteration, the algorithm is proved to be convergent.

Algorithm 1: Algorithm pseudo code

```
centroids = create Cent(dataSet,k) while cluster Changed:
    cluster Changed = False for i in range(m):
      minDist = inf; minIndex = -1 for j in range(k):
          calculate minDist, return minIndex if centroids changed:
        cluster Changed = True clusterAssment[i,:] = minIndex,minDist
    for cent in range(k):
      update centroids position(); return centroids, cluster Assment
```

The difference between those pseudo codes and the traditional K-means clustering is the choice of the initial clustering centers. In this paper, the selection of the initial clustering centers is optimized, the

ISSN: 2233-7857 IJFGCN

clustering results can be avoided to converge to local optimal solution effectively and the optimized algorithm accelerates the algorithm convergence speed

4.1. Rapid generation of Unity virtualscene

Firstly, the whole terrain is divided into n parts by grid processing, with its lower left corner coordinates to represent sub terrain. The Unity terrain is abstracted into a series of coordinate points, and the corresponding data sets are obtained.Secondly, data set is clustered by K-means algorithm, the value of k is 3. The K- means processing results are 3 text files, which store the coordinate points of different clusters

At last, text files obtained by K-means clustering algorithm are imported into Unity and the coordinate points of text files are read automatically by C# scripts. For the coordinate points, we should instantiate the corresponding models and generate Unity scenerapidly.Eachsubterraincorrespondstoonlyonemodeltheoretically,itcanalso be enriched in order to achieve a better visual effect. We can generate randomly m coordinate points in each sub terraint_{-*-}, a formal definition is asfollow:$\{(x_{**},y_{**})|x_{/}<x_{**}\leq x_{/}+1, y_{/}<y_{**}\leq y_{**}+1\}\{m = 1,2,3 \dots\}$, so we can add multiple models in each sub terrain. The basic steps of Unity scene rapid generation are as follows:

- Step1: Getting the Unity terrain data, terrain data is abstracted into initial data set.
- Step2: According to data set, the distribution function of initial clustering centers is fitted.
- Step3: Getting the initial clustering centers.
- Step4: Each object in the date set is assigned to the nearest cluster based on the dis- tances between itself and each clustering center.
- Step5: Determining whether clustering centers are changed. If true, repeat (3) ~ (5), else output clustering results.
- Step6: In Unity, we can read those clustering results and instantiate different models for each class.
- Step7: Checking the instantiated results and complete the rapid generation of Unity scene.

5. Experimental results andanalysis

The experimental data are derived from Unity terrain data, the DB index of clustering algorithm and the number of iterations for clustering algorithm is recorded. D Bind ex and the number of iterations are used to measure the clustering quality and convergence speed of the algorithm. The smaller the DB index is, the higher the clustering quality will be. Moreover the smaller the number of iterations is, the faster the convergence speed becomes. Experimental operating environment is Intel(R) $Xeon(R)$ $CPUE5-2650$ $v4$ @ $2.20GHz$, using python programming, the interpreter is python 2.7.11.

The K-means algorithm based on the Unity terrain data is used to cluster data sets. According to clustering results, different classes are corresponding to different models. We can run the program to complete the rapid generation of Unity scene. For each cell, we choose to instantiate one model and get a better visual effect. Specific results show inFig.3:

ISSN: 2233-7857 IJFGCN

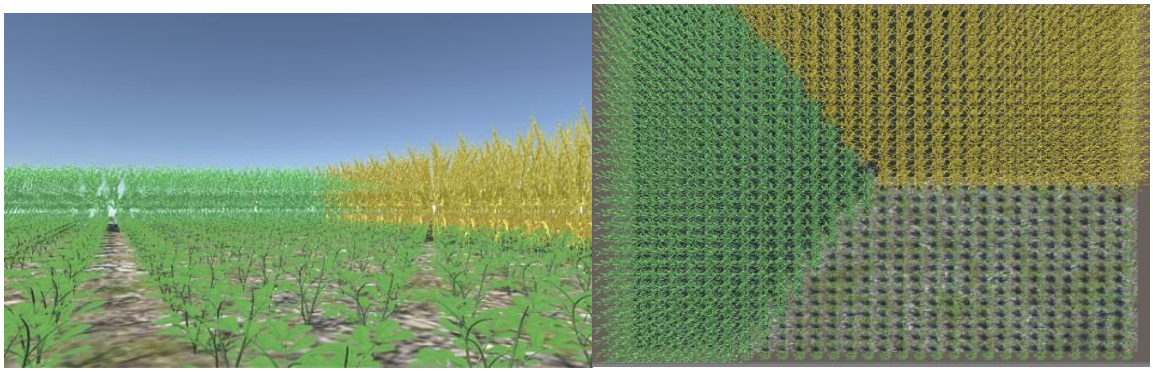

Fig 3. Unity scenes generate rapidly

According to data sets, the traditional K-means algorithm and the K-means algorithm based on Unity terrain data are compared, DBindex and the number of iterations are recorded when algorithm is convergent. For each data set, 50 experiments are run and clustering results are stored in text documents. Results are depicted in Table 2 and Table3.

Table2. Original clustering analysis

Data Set	maxIter	minIter	meanIter	maxDB	minDB	meanDB
D_{10}	10.0	4.0	6.0	1.25	1.21	1.24.
D_{20}	18.0	5.0	10.1	1.23	1.19	1.20
D_{30}	37.0	6.0	20.8	1.22	1.19	1.19
D_{40}	38.0	7.0	22.8	1.22	1.18	1.19
D_{50}	34.0	7.0	19.2	1.22	1.18	1.19

Table3. Clustering analysis for Unity terrain

k-means for unity terrain						
Data Set	maxIter	minIter	meanIter	maxDB	minDB	meanDB
D_{10}	8.0	3.0	5.6	1.25	1.21	1.23
D_{20}	19.0	4.0	9.2	1.23	1.19	1.21
D_{30}	31.0	7.0	18.2	1.22	1.19	1.19
D_{40}	42.0	7.0	22.2	1.19	1.18	1.19
D_{50}	32.0	10.0	17.5	1.22	1.18	1.19

Each column in the table are corresponds to the maximum number, the minimum and average values of iterations, and the maximum, minimum and mean values of the DBindex respectively. Each row in the table corresponds to a dataset, namely D_{10}, D_{20}, D_{30}, D_{40}, D_{50}. According to above tables, it is known that optimized K-means algorithm can effectively reduce the number of iterations required for convergence, when DB index is basically unchanged. For each data set, we also do the corresponding visualization (See Fig.4 and Fig.5).

ISSN: 2233-7857 IJFGCN

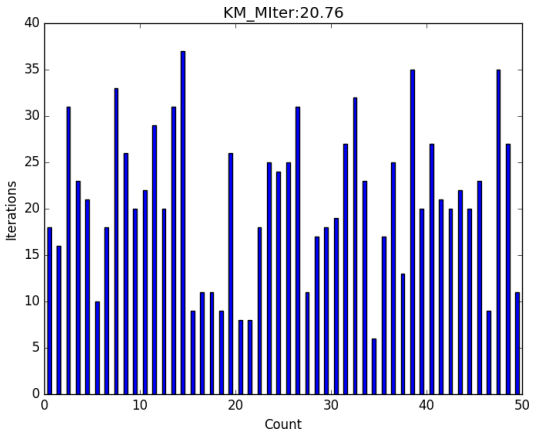

Fig.4 Original clustering analysis

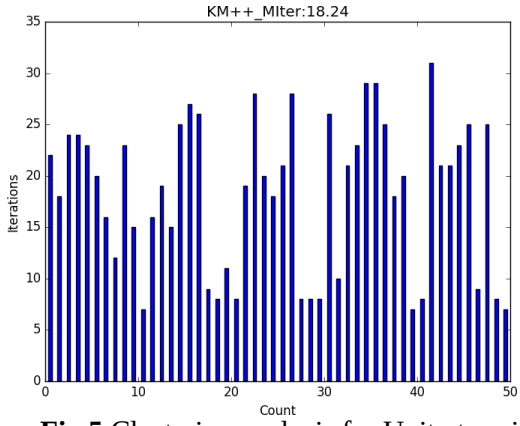

Fig.5 Clustering analysis for Unity terrain

The horizontal coordinates represent the number of algorithm execution and the vertical coordinates represent the number of iterations required for convergence, the title of graph is the average number of iterations. According to above results, the average number of iterations for K-means algorithm based on Unity terrain data is significantly lower than traditional K-means algorithm.

6. Conclusion

This paper mainly provided novel method for generating Unity scene rapidly by Machine Learning algorithm. According to Unity terrain data, data sets are obtained and then the initial clustering centers of K-means algorithm are determined. Data sets are processed by optimized K-means algorithm, according to the clustering results, corresponding models are instantiated in Unity. At end, the traditional K-means algorithm and K-means algorithm based on Unity terrain data are compared, and it also analyzes the clustering quality and the number of iteration required for convergence.

TheexperimentalresultsshowthattheK-meansclusteringalgorithmbasedonUnity terrain data can effectively reduce the number of iterations when the clustering quality is guaranteed, and the feasibility of using Machine Learning algorithm to realize the rapid generation of Unity scene isverified

References

[1] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(4): 463-484, 2012.

ISSN: 2233-7857 IJFGCN

[2] H. MacLeod, S. Yang, K. Oakes, K. Connelly, and S. Natarajan. Identifying rare diseases from behavioural data: A machine learning approach. In *CHASE*, 2016.

[3] N. MacLeod, M. Benfield, and P. Culverhouse. Time to automate identification. *Nature*, 467: 154-155, 2010.

[4] A V L N Sujith, Dr. A Rama Mohan Reddy, Dr. K Madhavi, "EGCOPRAS: QoS-Aware Hybrid MCDMModel for Cloud Service Selection inMulti-Cloud Environment" Journal of Adv. Research in Dynamical & Control Systems, Vol. 11, 06, 2019 (Scopus)

[5] S. J. Stolfo, D. W. Fan, W. Lee, A. L. Prodromidis, and P. K. Chan. Credit card fraud detection using meta-learning: Issues and initial results. In *AAAI*, 1997.

[6] F. S. Hanifah, H. Wijayanto, and A. Kurnia. SMOTE bagging algorithm for imbalanced dataset in logistic regression analysis. *Applied Mathematical Sciences*, 9(138): 6857-6865, 2015

[7] A V L N Sujith, Dr. A Rama Mohan Reddy, Dr. K Madhavi, "QoS-Driven Optimal Multi-Cloud ServiceComposition Using Discrete and FuzzyIntegrated Cuckoo Search Algorithm" International Journal of Engineering and Advanced Technology (IJEAT)ISSN: 2249-8958, Volume-8 Issue-5, June 2019

[8] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Doll´ar. Focal loss for dense object detection. In *IEEE ICCV*, 2017.

[9] Public available at the Kaggle platform: https://www.kaggle.com/c/imet-2019-fgvc6/data.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[11] S. Xie, R. Girshick, P. Dollr, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[12] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[13] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

ISSN: 2233-7857 IJFGCN