2021 20th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Parallelization implementation of topographic viewpoint filtering algorithm based on terrain viewshed using MPI and OpenMP

Yiwen Wang, Tao He, Wanfeng Dou
School of Computer & Elelctronic Information, Nanjing Normal University, Nanjing 210023
Nanjing 210023, Jiangsu, China
douwanfeng@njnu.edu.com

***Abstract*—The problem of multi-point visibility is an important part in terrain visibility analysis. It is widely used in military, urban planning, protection of endangered animal and other fields. Siting observer point is a kind of multi-point viewshed problem. It is generally abstracted as selecting the least number of viewpoints on a given terrain to maximize the joint viewshed covered by them. A candidate viewpoint filtering method proposed by Wang et al. to effectively solve this problem, includes k-means clustering of candidate viewpoints, calculation of contribution degree of each viewpoint, and ranking of viewpoints on contribution in each cluster. But this method is very time-consuming. Therefore, this paper adopts the MPI parallel program framework to implement the parallelization for the k-means algorithm, and uses OpenMP to realize the parallelization of the ranking process of view contribution of candidate viewpoints and the calculation process of view contribution of each viewpoint. The experimental results show that our parallelization scheme of viewpoint filtering can greatly reduce the calculation time and improve the efficiency of the algorithm.**

Keywords- multi-point viewshed analysis; observer siting; viewpoint filtering; parallelization

I. INTRODUCTION

Terrain visibility analysis uses computer graphics technology and computer geometry principle to solve the problem of visibility between target point set and observation point set on terrain. It is an important part of geospatial analysis [1]. Visibility issues can generally be divided into two categories: The first category is the calculation of visibility information of a given monitoring point. This information mainly includes the visual size and range of monitoring points of different properties and types[2]. The second category is to use these visual information to solve different application problems, such as location planning, path planning, real-time roaming and others[3].

Visibility analysis has been widely used in the process of environmental resource management, urban planning and military activity analysis, such as optimal path planning[4] , forest fire monitoring[5], communication tower positioning[6], military observation points siting[14],etc. These applications can be attributed to multi-point viewshed analysis. However, unlike single-point viewshed analysis that only needs to calculate the viewshed of a single point, multi-point viewshed analysis needs to obtain the largest or smallest joint viewshed by calculating the viewshed of a set of points. Therefore, the problem of observation point siting based on multi-point viewshed analysis is a combinatorial optimization problem, because it involves the optimal combination of multiple viewpoints and multiple target points, so it is also an NP problem[7]. The complexity of solving such a problem will increase exponentially with the increase of the number of candidate viewpoints.

At present, the viewpoint filtering method can effectively solve the multi-viewpoint viewshed. It first obtains some feature points on the terrain as candidate viewpoints, then evaluates these candidate points, and filters out some viewpoints with low contribution or poor quality until the required number of viewpoints is met[12-13]. In the problem of siting observation points, it is necessary to calculate the viewshed of a large number of terrain points, which is very time-consuming. Even if the terrain feature points are selected as the candidate viewpoints, the number is still very large. Therefore, it is of great significance to study the parallelization method and its solution of multi-viewpoint viewshed analysis. However, the current research mainly focuses on the parallelization of single-point viewshed computation, such as the parallelization of X-draw algorithm[14]. In the fast filtering method of terrain point, before performing the filtering operation, all the candidate points should be clustered by k-means method[15] , and then the terrain points in each cluster are sorted according to their viewshed quality. Although the feature points extracted from the whole terrain are used as candidate points, the number of these candidate observation points is still very large. Therefore, clustering operation and visual quality calculation of each terrain point takes a lot of time. This paper considers the parallelization of each step in the filtering method to improve the computational efficiency.

II. TERRAIN POINT FILTERING METHOD AND ITS FRAMEWORK OF PARALLEL METHOD

A. Terrain point filtering method

The main idea of the fast filtering method for candidate viewpoints is given as follows [12-13]. The candidate viewpoints are clustered by K-means algorithm based on geometric distance, and than N clusters are obtaind. Then, the viewpoints in each cluster are sorted according to their viewshed quality. Then the viewpoints with the lowest viewing quality in all clusters are filtered cyclically until the number of viewpoints in all cluster meet the predetermined number requirements. The fast filtering method for candidate viewpoints includes the following steps:

1)Topographic feature point extraction. With the acquisition of high precision terrain data, the number of terrain grid points is also increasing. If every point on the entire terrain is treated as a candidate viewpoint, it will inevitably lead to a large amount of time overhead. Therefore, using topographic feature points as candidate viewpoints can greatly reduce the time cost of location optimization calculation. Available topographic features

2473-3636/21/$31.00 ©2021 IEEE
DOI 10.1109/DCABES52998.2021.00037

include mountaintops, ridge points, saddle points, flat locations[16],etc. The research shows that the topographic feature points have higher visibility than the general grid points in the terrain.

2)Clustering all topographic feature points. According to the specific requirements of the actual number of site selection plans, we can determine the total number N of clusters. In this paper, K-means clustering algorithm is used to cluster the terrain feature points, and the clustering centers of multiple clusters and each cluster are obtained.

3)Sorting the terrain feature points by visual quality in each cluster. The evaluation index is calculated according to the viewshed quality of the viewpoints in each cluster and sorted according to the value. Here, all viewpoints in clusters are sorted in descending order of their evaluation index.

4)Filter the feature points with the lowest quality of viewshed. Comparison is made on the viewshed quality of feature points in all clusters. If a feature point in a cluster is with the lowest viewshed quality in all clusters, this point will be deleted. Otherwise, it will be inserted into another cluster that is not with the smallest view quality.

5)Repeat steps 3-4 until the numbers of points in each cluster meets the expected requirements.

The fast filtering method of candidate feature points traverses all clusters until the number of viewpoints in the cluster satisfies the termination condition. Please pay attention to that when repeating Step 3, just reassess the viewshed quality of all feature points in the cluster where new feature points are inserted, rather than all clusters.

B. Parallel framework of topographic point filtering algorithm

The terrain point filtering algorithm is divided into three parts: the clustering of terrain points, the sorting of terrain points in the cluster, and the filtering of terrain points.

Due to the large number of the terrain points, even if the terrain feature points are extracted as candidate points, the number of these candidate points is still very large. The use of serial k-means clustering method will cause the problem of low computational efficiency. Therefore, MPI parallel k-means algorithm is considered. When the clustering operation is completed, the viewpoints in each cluster should be sorted. When evaluating the quality of viewpoints, the sights of each viewpoint should be calculated. The calculation of the sights of a single viewpoint is time-consuming. Therefore, when calculating the viewshed of a large number of viewpoints, we adopt the OpenMP parallel program framework to reduce the calculation time. The last part is the filtering of candidate viewpoints. Considering that the viewpoints in the cluster needs to be re-sorted according to the quality of their viewing areas during the filtering process, OpenMP is also used for parallel calculation to improve the computational efficiency.

In summary, the parallelization method based on terrain point filtering algorithm is mainly divided into the following three parts: 1)parallelization of k-means clustering based on MPI; 2)parallelization of terrain point sorting based on OpenMP; and 3)parallelization of terrain point filtering based on OpenMP.

III. PARALLEL METHOD OF TERRAIN POINT FILTERING

A. Parallelization of k-means Clustering

MPI(Message Passing Interface) is one of the important technologies used in the cluster. It is an application program interface based on message passing model. The main feature is that between different processors, it uses the network to transfer messages to achieve mutual communication, and completes the synchronization between tasks. Message passing methods is generally used in distributed storage structure. Because the number of terrain points is very large, the original K-means clustering algorithm needs to spend a lot of time to calculate the Euclidean distance between each data point. After research, it can be found that the distance between each terrain point and the obtained terrain point as the clustering center is independent of each other and does not interfere with each other, so the algorithm can be calculated in parallel. Therefore, this paper first divides the original data into multiple sub-data blocks, and then allocates each sub-data block to an independent processor, and each processor only needs to calculate the distance between the topographic point in the assigned sub-data block and each cluster center separately, and then the clustering result of the sub-data block processed by each processor can be obtained. Finally, the global clustering results are obtained by calculating the output results of each processor through a method similar to the Reduce function in MapReduce framework. The proposal of the implementation of the parallelization of k-means clustering algorithm based on MPI is mainly to use master-slave mode. The task of the master node is to divide and distribute data. The slave nodes are allocated to with the local data to complete the calculation task and the results are fed back to the master node. The process of parallelization is as follows:

1)Assuming that the master node is process 0, the process reads data from the file and allocates the data to other processes.

2)The main process selects each cluster center and sends it to other processes.

3)Other processes calculate the distance between each point in the assigned data and the cluster center point, mark the category of each point, and then calculate the sum of the distances between each terrain point in each class and the cluster center, and finally return the results to the main process.

4)Process 0 calculates the new center point and sends it to other processes, and calculates the sum of distances between all terrain points in the class from other processes and its center point.

5)Repeat step 3 and 4 until the sum of the distances of all clusters in step 4 or the cluster center no longer change (i.e. convergence).

B. Parallelization of terrain viewpoint sorting

OpenMP is a thread-based programming framework, which is generally used in the parallel system of shared

memory. For a parallel program based on multithreading, it is necessary to block and schedule the loops in the program to better achieve load balancing, and ensure that the CPU utilization is the highest and has been in a busy state during most of the running time as long as possible. At the same time, it minimizes various costs, such as scheduling overhead, synchronization overhead and the switching overhead between contexts, so as to optimize performance. The parallelization of terrain point sorting in each cluster adopts the static balanced scheduling strategy. This part is to sort the quality of terrain points in each cluster. Each cluster does not interfere with each other, so the pragma instruction can be used to parallelize the sorting process. In the OpenMP-based terrain point sorting method, only clusters are partitioned. Each cluster is processed by a single thread, and there is no interference between clusters, so there is no shared variable. Therefore, the pseudo code of the parallel algorithm for sorting terrain point based on OpenMP is shown as follows.

```
input K clusters
#pragma omp parallel for num_threads(n)
(n is the specified number of threads)
for i from 0 to (k-1)
{
compute viewshed quality of every viewpoint;
sort these viewpoints according to their viewshed quality;
}
```

C. *Parallelization of terrain point filtering*

In the process of terrain point filtering, the terrain points in each filtering cluster may be moved to adjacent clusters, and each cluster has strong correlation and poor parallelism. When filtering terrain points, it is necessary to calculate the viewshed quality of the viewpoint in the adjacent clusters, which is also the most time-consuming part in the whole filtering process. Therefore, this part can use OpenMP to improve efficiency.

The calculation of viewshed quality involves the view and view repetition of a single viewpoint. When calculating the number of view repetitions, it is necessary to calculate the joint view of multiple viewpoints. The essence of this series of calculations is to calculate the viewshed of a single viewpoint, and the viewshed of a single viewpoint does not affect each other. Therefore, after calculating the single point viewshed with parallel computing mode, the main thread can calculate the viewshed quality.

First, the parallel for statement is placed before calculating the for loop in each view field. Parallel is used in combination with the for instruction, representing multiple threads to execute the code in the for loop in parallel. It also has two functions: generating parallel domains and assigning tasks. Since the viewpoint quality needs to calculate the joint view, and every time a field of view is calculated, the joint field of viewshed array V must be changed at the same time. Therefore, in the case of multi-threaded parallelism, the array V is a shared variable. Where the shared clause in OpenMP is used to modify V, the role of shared clause is to specify one or more variables as shared variables between multiple threads. Finally, after each thread completes the execution task, the viewpoint quality formula is usd to calculate its quality, because the time for each thread to complete the task will be different, and the barrier is generally used to synchronize the threads executing the code in the parallel region. When each thread reaches the barrier instruction, it needs to stop and wait until all other threads execute to the barrier to continue to execute the remaining part. Therefore, adding the #pragma omp barrier statement can realize thread synchronization. The pseudo code of the parallel calculation of viewpoint quality in the filtering process is shown as follows.

```
Input: cluster C; single viewpoint p; Initialize the joint view array V
# pragma omp parallel for shared（V）
for i from 1 to N (N is the number of viewpoints in cluster C)
{
 calculate the viewpoint of i;
 modify array V from the perspective of view i；
 #pragma omp barrier
}
compute viewshed Vp of viewpoint P;
compute the overage ratio O according to Vp and shared viewshed array V;
compute viewshed quality of viewpoint P according to O and Vp of P.
```

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The experiment compares the serial method with the parallelization method, and verifies the computational efficiency of the three parallel methods.

1)Experiment with k-means based on MPI

The experimental environment is two computers with CPU@2.0GHz, and 4GB memory, and windows 10 system. A total of eight processes for the program are started to run. The data adopts 1000*1000 digital elevation model terrain with a resolution of 5 meters. In order to compare, 10000, 100000, and 1000000 data points were tested. The running time is shown in Table I.

Experimental result shows that the serial execution of K-means algorithm takes the longest time. After paralleling it with the MPI framework, the speed of data processing is significantly improved. With the continuous increase of the amount of data, the running time of serial algorithm rises sharply, but it also reflects the advantage of parallel computing. When the data sizes up to 10000, 100000 and 1000000, the running time of k-means based on MPI is about 1 / 2,1 / 10 and 1 / 433 of that of the serial k-means algorithm, and the speedups ratio reaches 2.0, 10.45 and 433.49 respectively. Thus, the efficiency of parallel computing algorithm proves to be greatly improved.

TABLE I. Comparison of running time and accelerate rate

Data size	10000	100000	1000000
k-means	149ms	7982ms	3652168ms
k-means based on MPI	75ms	764ms	8425ms
Accelerate rate	2.0	10.45	433.49

2)Experiment of terrain point sorting algorithm based on OpenMP

Since the OpenMP parallel program design framework is mainly to make full use of the CPU to accelerate the calculation, the experiment is carried out on a computer

with CPU@2.0GHz and 4GB memory, and windows 10 system. 1000 selected terrain feature points are clustered into 35, 45 and 65 classes as the experimental data. The number of threads can be specified with num_threads in OpenMP. The number of threads set in this experiment is 2 and 4, respectively. Table II shows the resulting experimental running time of 35, 45 and 65 classes sorted by serial and parallel methods.

It can be seen from table 2 that the use OpenMP to parallelize the sorting of terrain points in the cluster based on visual quality Index can improve the computational efficiency. Sorting viewpoints in each cluster requires many times of computations of the visibility of each viewpoint in the cluster, so serial sorting takes a long time. When sorting viewpoints in35, 45 and 65 clusters respectively, the computation time of serial algorithm is about 1.48, 1.52 and 1.55 times that of parallel computing with two threads, and 2.69, 2.91 and 2.8 times that of parallel computing with four threads. It can be seen that the utilization of the CPU is significantly increased when multithread parallel computing is enabled.

TABLE II. Comparison of running time

number of clusters	35	45	65
Serial algorithm	682s	649s	608s
Parallel algorithm with two threads	458s	425s	391s
Parallel algorithm with four threads	253s	223s	217s

3)Experiment on Topographic Point Filtering Algorithm Based on OpenMP

The experiment is also carried out on a computer with CPU@2.0GHz and 4GB memory, and windows 10 system. The experimental data are 35, 45 and 65 classes that originate from terrain feature points on D1 terrain after clustering and sorting operations applied. The number of threads is also 2 and 4. Table III shows the running time of serial candidate viewpoint filtering(CVF) and parallel CVF algorithms. It can be seen from Table III that using OpenMP to filter terrain points in parallel can significantly shorten the calculation time. When filtering terrain points, it takes the longest time to calculate the quality of viewpoints. In this experiment, 35, 45 and 65 clusters are filtered respectively. From the running time results, it is concluded that the time of serial computing is about 1.55, 1.27, 1.69 times of the parallel computing time when two threads are used, and 2.81, 2.8, 2.75 times of the computing time when using four threads.

TABLE III. Running time of serial CVF and parallel CVF

cluster number	35	45	65
serial CVF	45s	14s	22s
parallel CVF with two threads	29s	11s	13s
Parallel CVF with four threads	16s	5s	8s

V. CONCLUSION

In this paper, the rapid filtering method of candidate feature points with multiple viewpoints is parallelized step by step. The k-means algorithm is mainly parallelized by MPI framework, and using OpenMP is used to parallelize the sorting of terrain points in each cluster and the final terrain point filtering. Experiments show that the partial parallelization of computing time can significantly shorten the computing time and improve the computational efficiency. The parallelization method is carried out step by step, which is only suitable for candidate terrain point filtering method. The future research direction is to find a parallelization solution to make it suitable for various problems of multi-viewpoint visibility.

FUNDING

This work is supported by the National Natural Science Foundation of China [grant number 41771411].

REFERENCES

[1] Fisher P F. First Experiments in Viewshed Uncertainty: Simulating Fuzzy Viewsheds[J]. Photogrammetric Engineering and Remote Sensing, 1992, 58(3): 345-352.

[2] Pin L. Terrain Visibility Analysis Based on Line of Sight[J]. Computer Engineering and Applications, 2006, 42(8):223-226

[3] LEE J. Analyses of visibility sites on topographic surfaces[J]. International journal of geographical information systems, 1991, 5(4):413-429.

[4] Franklin, W.R., et al. Slope Accuracy and Path Planning on Compressed Terrain in Headway in Spatial Data Handling[C]. 13th International Symposium on Spatial Data Handling, Montpellier, France, 23-25 July, 2008.

[5] Bao S., Xiao N., Lai Z., Zhang H., & Kim C. Optimizing watchtower locations for forest fire monitoring using location models[J]. Fire Safety Journal, 2015, 71: 100-109.

[6] Akella M. R., Delmelle E. M., Batta R., Rogerson P. A., and Blatt A.. Adaptive Cell Tower Location Using Geostatistics[J]. Geographical Analysis, 2010, 42(3):227-244.

[7] Kammer F., Löffler M., Mutser P., et al. Practical Approaches to Partially Guarding a Polyhedral Terrain[J]. Geographic Information Science, 2014, 8728.

[8] Puppo E., & Marzano P. Discrete visibility problems and graph algorithms[J]. International Journal of Geographical Information Systems, 1997, 11(2), 139-161.

[9] Franklin W. R., Ray C. K., and Shashank M. Geometric Algorithms for Siting of Air Defense Missile Batteries[OL]. Research Project for Battle 2756, 1994. http://www. ecse. rpi. edu/Homepages/wrf/research/p/tec_report.pdf.

[10] Sorensen P. A. & Lanter D. Two algorithms for determining partial visibility and reducing data structure induced error in viewshed analysis[J]. Photogrammetric Engineering and Remote Sensing.1993,59. 1149-1160.

[11] Wang J., Robinson G. J., White K. A Fast Solution to Local Viewshed Computation Using Grid-Based Digital Elevation Models[J]. Photogrammetric Engineering and Remote Sensing, 1996, 62(10):1157-1164.

[12] Yu T., Xiong L., Gao M., Wang Z., et al. A new algorithm based on Region Partitioning for Filtering candidate viewpoints of a multiple viewshed[J]. International Journal of Geographical Information Science, 2016, 30(11), 2171-2187.

[13] Wang Y. W. & Dou W. F. A fast candidate viewpoints filtering algorithm for multiple viewshed site planning[J], International Journal of Geographical Information Science, 2020, 34(3): 448-463.

[14] Wu Y. L. An algorithm for computing viewsheds based on reference planes[J]. Wtusm Bulletin of Science and Technology, 2001, 1 (6):19-21.

[15] Kanungo T., Mount D. M., Netanyahu N. S., Piatko C. D., Silverman R., & Wu A. Y. An efficient k-means clustering algorithm: analysis and implementation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002, 24(7):881-892.

[16] Kim Y. H., Rana S., & Wise S. Exploring multiple viewshed analysis using terrain features and optimization techniques[J]. Computers & Geosciences, 2004, 30(9-10), 1019-1032.