2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

A fast shortest path method based on multi-resolution raster model

Qiuling Tang
Dept. of Computer Science & Technology
Nanjing Normal University
Nanjing, China
182202019@stu.njnu.edu.cn

Wanfeng Dou
Dept. of Computer Science & Technology
Nanjing Normal University
Nanjing, China
douwanfeng@njnu.edu.cn

***Abstract*—The shortest path problem based on spatial geography is a research hotspot in the intersection of GIS and path planning. In order to improve the efficiency of solving the shortest path based on high-precision terrain data, this paper proposes a fast spatial terrain shortest path based on multi-resolution DEM data. This method uses different multi-resolution DEM raster data models to represent the same terrain. First, the coarse-grained shortest path is solved on the low-resolution terrain, then the shortest path between adjacent grids is solved in parallel on the high-resolution terrain data according to the corresponding mapping mode. This method reduces the search range between grids and greatly improves the efficiency of solving the shortest path. Experimental results show that the method proposed in this paper can not only solve the optimal path on the DEM raster data structure, but also improve the efficiency of directly solving the shortest path on a large-scale high-resolution terrain.**

Keywords- spatial geography; shortest path; multi-resolution DEM; A* Algorithm

I. INTRODUCTION

The shortest path problem is a classic problem in graph theory, which means finding a path with the shortest distance between two nodes. Existing methods to accelerate the calculation of the shortest path are to abstract the path problem into a graph and perform the shortest path calculation on the converted graph. Such methods are called hierarchical or target-oriented methods. The idea of the hierarchical method is to convert a graph into a multi-layer coverage map [1] in advance to quickly obtain the shortest path, such as the highway multi-level method [2], highway node path planning method [3], etc. The target-oriented method preferentially considers the edge that can shorten the distance to the target point by removing the edge that is not on the shortest path, such as the ALT (A* search, Landmarks and Triangle inequality) method [4], Arc-Flags method [5], etc.

With the increasing range of DEM (Digital Elevation Model) [6-7] terrain, the amount of data increases geometrically, and the complexity of the path search algorithm on the raster terrain will also increase exponentially [8]. Therefore, solving the optimal path on the high-resolution DEM data is quite time-consuming. For massive data or computationally intensive problems in path search, many domestic and foreign scholars will use parallel computing to improve the optimal path solution efficiency. Djidjev performs subgraph decomposition on undirected graphs or directed graphs, and then uses GPUs (Graphics Processing Units) to solve APSP (All-Pairs Shortest-Path) problems [9]. Madduri and others chose the Cray MTA-2 multi-threaded parallel architecture to solve the shortest path problem on large-scale data sets [10], which has the features of fine-grained parallelism and low-overhead synchronization primitives.

II. MATERIALS AND MODEL

A. Materials

The terrain data in this study selected three typical terrains of plains, hilly areas and mountains, which are widely representative, to ensure that the analysis results are universal. The experimental materials is composed of three DEMs from ASTER GDEMV2 with 30m grid spacing, with TIFF format and serial numbers of N32E120, N31E116 and N28E97 respectively. The algorithm in this study and other comparison algorithms will perform path analysis based on these materials. The locations and terrain characteristics of the three DEMs are shown in Table I.

TABLE I. DISPLAY OF EXPERIMENTAL MATERIALS(DEMS)

DEM	Location	Area	Terrain Type
N32E120	E120 -121 , N32 -33	Middle and Lower Yangtze River Plain	Plain
N31E116	E116 -117 , N31 -32	Southern Anhui, China	Hilly
N28E097	E97 -98 , N28 -29	Sichuan, China	Mountains

B. The construction of terrain hierarchical model

First, the DEM data of the original terrain is obtained to build a terrain hierarchy model with different resolutions. According to the selected mapping method, the mapping from high-resolution original terrain to multi-layer low-resolution terrain is established, as shown in Figure 1.

We regard the topographic data with the highest resolution as the topographic representation of the highest layer in Figure 1, which is set to the 0th layer, and establish the corresponding DEM grid. The low-level ground resolution grids can be obtained by high-level high-resolution grid mapping, and there is a corresponding mapping relationship between the grids of different layers. For the mapping relationship, this paper chooses 2×2 and 3×3 mapping modes.

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00074

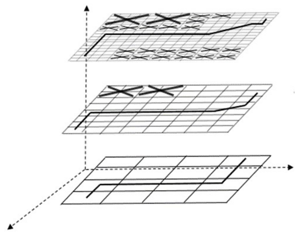

Figure 1. Terrain hierarchical model with different resolutions.

The element values stored in the raster terrain are elevation values. When we choose 2×2 mapping mode to map from high-resolution terrain to low-resolution terrain model, we calculate the elevation value of the grid due to no direct corresponding grid. As shown in Figure 2, for the four grids A, B, C, and D in the high-resolution 2×2 area, we use interpolation to get the elevation value of the next grid S. When selecting the 3×3 mapping mode, we directly select the grid at the center of the high-level 3×3 area as the center of the low-level grid. The terrain resolution ratios of the three resolutions constructed by 2×2, 3×3 mapping modes are 1:4:16, 1:9:81 respectively.

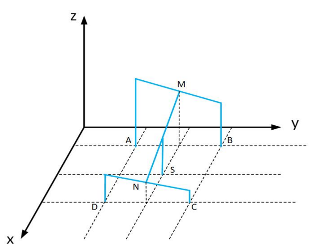

Figure 2. 2×2 interpolation method.

III. SOLVE THE SHORTEST PATH ON LOW-RESOLUTION TERRAIN

After constructing a terrain hierarchical model with different resolutions, we solve the shortest path on the lowest resolution terrain data. Considering the influence of geospatial elevation, slope, and obstacles, the A* algorithm is adopted to find the optimal path.

A. Slope pruning

DEM raster data is represented as graph structure data. The graph structure transformed in this paper is directed graph. The points in the graph are the points in the grid where elevation values are stored. The connection between points is measured by the steepness of the terrain surface, that is, the slope between the two points needs to be calculated. The slope is used as the basis for judging whether the connection between the vertices in the graph is pruned. We need to select a suitable slope value for pruning according to the terrain. If it is greater than this slope value, we choose to delete the edge of the vertex.

Here, the degree representation method is applied to calculate the slope. The edges between adjacent grids on the terrain are along the X-axis or Y-axis or diagonal direction. Under the eight-neighborhood model of vertices, the calculation method is shown in equation (1).

$$\tan\theta = E_d / DL \tag{1}$$

Where, E_d is the difference in elevation value, DL is the plane distance in the X-axis or Y-axis or diagonal direction, and θ is the slope between two adjacent grid points on the terrain.

B. Pruning slope attenuation

Existing studies have shown that DEM resolution will greatly affect the average slope of the terrain. The average slope of the ground has a strong linear relationship with the DEM resolution, and the average slope will decrease as the resolution decreases. Therefore, in the terrain hierarchy model of this paper, it is necessary to select appropriate slope values for each layer of terrain for pruning. We use the linear relationship between terrain resolution and average slope to attenuate the selected pruning slope of each layer.

As shown in Figure 3, taking the 2×2 mapping mode as an example to show the linear fit of its resolution and average slope. In the 2×2 mapping mode, the linear fitting of the resolution and average slope of the plains and hills is more significant, and the mountain range error is a bit larger. The decline of the average slope is positively related to the terrain complexity. The 3×3 mapping mode has the same effect. Here we uniformly regard the average slope and resolution as the mode of $y = ax + b$, where y is the average ground slope and x is the DEM resolution. The values of the coefficients a and b are related to the topography.

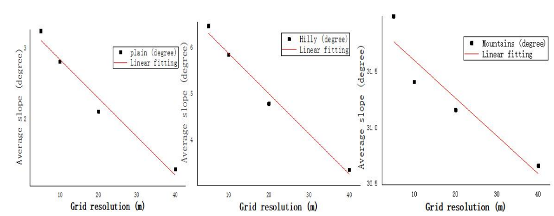

Figure 3. Linear fitting of the resolution and average slope of plains, hills and mountains in the 2×2 mapping mode.

Through the above linear fitting, We also assume that the pruning slope and DEM resolution of each layer also have the same linear relationship, following the model of $y = ax + b$, where y is the pruning slope and x is the DEM resolution. The value a in $y = ax + b$ is kept unchanged, the value b can be calculated by the pruning slope value and the terrain resolution of current terrain layer. Then, the pruning slope value of the lower terrain layer is calculated according to the corresponding terrain resolution.

C. Obstacles

Considering that there will be obstacles in the terrain, such as swamps and rivers, that hinder progress, we mark these obstacles. After solving the optimal path on the lowest resolution terrain, if the node of path is an obstacle, the entire area on the upper layer will be an obstacle.

IV. PATH GRID INVERSE MAPPING BETWEEN HIGH AND LOW RESOLUTION TERRAIN

Since each grid on the path corresponds to multiple grids on the high-resolution terrain, we map each grid of the path on the low-resolution terrain to corresponding grid on high-resolution terrain. For the 2×2 mapping mode, the grid mapping relationship is shown in Figure 4. The blue grid points in Figure 4 (a) are the path nodes in the optimal path solved on the low-resolution terrain. Grid 0 is the starting point, grid 3 is the target point, and the grid sequence of the planned route is 0-2-3. We give the rules applicable to the 2×2 mapping mode:

- If it is a route segment along the X axis, the orientation is shifted to the right to the corresponding grid.
- If it is a route segment along the Y axis direction, the orientation is offset to the corresponding grid.
- If it is a route segment in the diagonal direction (non-X axis and non-Y axis), take the grid where the route segment intersects.
- If it is the starting grid, it can be directly mapped to the corresponding starting grid.
- If it is a destination grid, it can be directly mapped to the corresponding destination grid.

Based on the above rules, the starting grid point 0 in Figure 4 (a) is mapped to the 00 area in Figure 4 (b). Similarly, the target point 3 is mapped to the grid 33. The grid point 2 in Figure 4 (a) is mapped to the grid point 23 in Figure 4 (b). In addition, for the 3×3 mapping mode, the rule of mapping low resolution to higher resolution is to directly map to the corresponding high-resolution center grid.

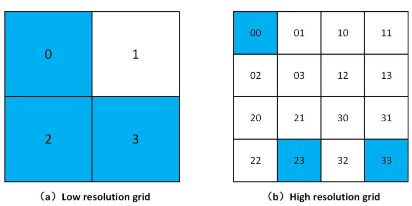

Figure 4. The mapping from low-resolution grid to high-resolution grid in the 2×2 mapping mode.

V. Parallel Solution of Local Paths Between Adjacent Grids on High-Resolution Terrain

After the path node mapping between high and low resolution terrain described above, the local path is solved on the high-resolution terrain. The local path solution is divided into two steps. The first step is to determine the search area between adjacent grids. The second step is to solve the shortest path between adjacent grid points in parallel.

A. *Determine the search area between adjacent grids*

The shortest path obtained on low-resolution terrain is a series of grids, and these grids are an area block in high-resolution terrain. Since the path between adjacent grids is not the final target path, the shortest path needs to be solved again for the areas. When planning each local path, the key task is to determine the search area between the pair of starting grids and the target grid. Since small search area may result in no feasible optimal path, the larger search area is used to solve the local shortest path.

We propose a simple method that uses the distance between the center of the starting grid and the target grid as the diameter to make a circle, and the area composed of the grids in the circle and on the circumference is the target search area. As shown in Figure 5, assuming that grid cells 00 and 32 are the starting grid S and the target grid D of a segment, respectively, the area covered by the circle obtained with their diameter (including the boundary) is the target search for the path to be sought region. The grid covered by the search area here includes 00, 01, 10, 02, 03, 12, 13, 20, 21, 30, 31, 22, 23 and 32.

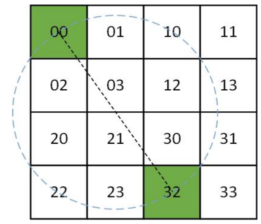

Figure 5. Determination of search area on high-resolution terrain.

B. *Local shortest path parallel computing method*

In this article, we use the mainstream parallel computing framework OpenMP, a library that manages multi-threaded levels, using the Fork/Join execution model. Fork is to create a new thread or wake up the created thread, and Join is a summary of the execution results of multiple threads. OpenMP reduces the difficulty of parallel programming. Developers do not need to pay attention to the details of parallel design, but only focus on business logic to greatly improve development efficiency.

VI. Fastest Shortest Path Algorithm Framework on Multi-Resolution DEM Terrain

This section introduces the specific process of fast shortest path algorithm based on multi-resolution grid model, and the flow chart is shown in Figure 6.

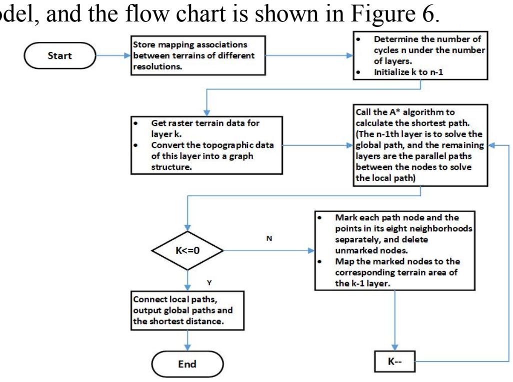

Figure 6. Flow chart of fast shortest path algorithm based on multi-resolution grid model.

VII. Experimental Results and Conclusion

This section is divided into two experiments. Experiment 1 compares the efficiency and accuracy of the serial algorithms in the 2×2 mapping mode and the 3×3 mapping mode. The serial algorithm is to select A* algorithm with model (i.e. MA*) and A* algorithm without model. Experiment 2 is the efficiency analysis of the algorithm parallelization in this paper.

A. *Serial Algorithm Experiment*

The experimental programming languages are all C++ languages. Table II below is the experimental data of different terrains in 2×2 and 3×3 mapping modes.

TABLE II. COMPARATIVE ANALYSIS OF ALGORITHMS ON DIFFERENT TERRAIN TYPES IN 2×2 AND 3×3 MAPPING MODE

			Plain	Hilly	Mountains
2×2	A*	Time/s	423.5	215.4	117.5
		Distance/m	2865.6	3315.4	2838.3
	MA*	Time/s	65.3	86.4	100.3
		Distance/m	3550.9	3951.5	2846.2
3×3	A*	Time/s	338.6	38.2	165.0
		Distance/m	2577.8	3428.7	2544.1
	MA*	Time/s	33.3	20.7	121.5
		Distance/m	2538.5	2737.6	2589.6

A* algorithm with model is simply referred to as MA*.

Figure 7 and Figure 8 below is a visualization of the data in Table II. (a) and (b) in the two figures represent time and path length comparison, respectively. This algorithm improves efficiency and is not the best path in terms of path distance. Moreover, the algorithm is more suitable for path planning in plain and hilly areas. Due to the complexity of the mountains, the algorithm of this paper is not notable in terms of efficiency and accuracy.

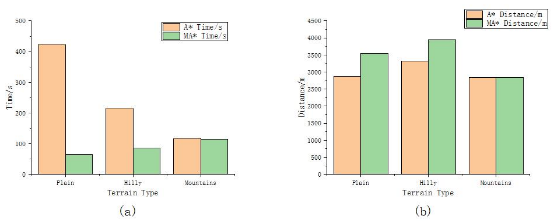

Figure 7. Comparison of algorithm efficiency and path distance between different terrains in 2×2 mapping mode.

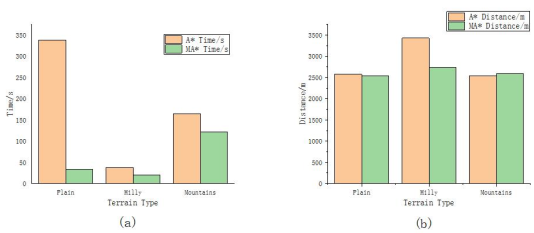

Figure 8. Comparison of algorithm efficiency and path distance between different terrains in 3×3 mapping mode.

Figures 9 and 10 show the resulting path diagrams in 2*2 and 3*3 mapping modes, respectively. The path in 2×2 mapping mode is more tortuous than the path in 3×3. We suspect this is related to interpolation and mapping rules.

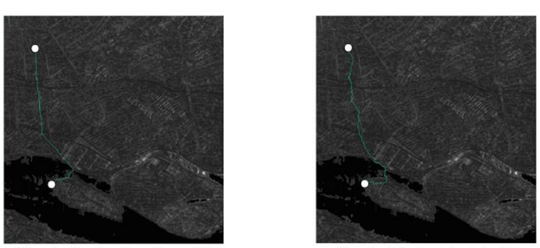

Figure 9. Shortest path display in 2×2 mapping mode.

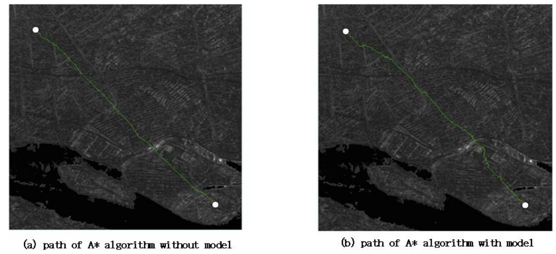

Figure 10. Shortest path display in 3×3 mapping mode.

B. *Comparison of serial and parallel time of model algorithm in this paper*

The experiment is conducted on a computer with a CPU frequency of 2.0 GHz, 8 GB of memory, and a Windows 10 system. Figure 11 is the parallelization of the MA* algorithm. The algorithm time decreases as the number of threads increases.

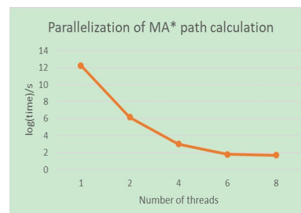

Figure 11. Parallelization of path calculation under the same range of terrain.

ACKNOWLEDGMENT

This work was partly supported by the National Natural Science Foundation of China (No. 41771411).

REFERENCES

[1] Bast H, Funke S, Sanders P, et al. Fast Routing in Road Networks with Transit Nodes[J]. Science, 2007, 316(5824): 566-566.

[2] Holzer M, Schulz F, Wagner D, et al. Engineering multi-level overlay graphs for shortest-path queries[C]. algorithm engineering and experimentation, 2006: 156-170.

[3] Sanders P, Schultes D. Engineering highway hierarchies[J]. ACM Journal of Experimental Algorithms, 2012, 17(1).

[4] Goldberg A , Kaplan H , Werneck R . Reach for A*: shortest path algorithms with preprocessing[M]// The Shortest Path Problem. 2009.

[5] Schilling, H., Wagner, D., and Willhalm, T., 2005. Partitioning graphs to speed up Dijkstra's algorithm. In: Proceedings of International Workshop on Experimental and Efficient Algorithms, Berlin, Heidelberg: Springer Press, pp. 189–202.

[6] Lakshmi S E, Yarrakula K., 2018. Review and critical analysis on digital elevation models. Geofizika, 35(2): 129-157.

[7] Fellegara R, Iuricich F, De Floriani L., 2017. Efficient representation and analysis of triangulated terrains. In: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. Pp. 1-4.

[8] Choi Y, Um J G, Park M H., 2014. Finding least-cost paths across a continuous raster surface with discrete vector networks. Cartography and Geographic Information Science, 41(1): 75-85.

[9] Djidjev H., Chapuis G., Andonov R., et al. 2015. All-Pairs Shortest Path algorithms for planar graph for GPU-accelerated clusters. Journal of Parallel and Distributed Computing.

[10] Madduri K, Bader D A, Berry J W, et al., 2006. Parallel shortest path algorithms for solving large-scale instances. Dimacs Implementation Challenge – the Shortest Path Problem, 74:249-290.