

## Article

# An Effective Method for Computing the Least-Cost Path Using a Multi-Resolution Raster Cost Surface Model

Qiuling Tang  and Wanfeng Dou \*

School of Computer and Electronic Information/School of Artificial Intelligence, Nanjing Normal University, Nanjing 210023, China; 202243007@njnu.edu.cn

\* Correspondence: douwanfeng@njnu.edu.cn; Tel.: +86-13770504265

**Abstract:** Calculating the least-cost path (LCP) is a fundamental operation in raster-based geographic information systems (GIS). The LCP is applied to raster cost surfaces, in which it determines the most cost-effective path. Increasing the raster resolution results in a longer computation time to obtain LCP. This paper proposes a method for calculating the LCP using a multi-resolution raster cost surface model to enhance computational performance for large-scale grids. The original raster cost surface is progressively downsampled to generate grids of decreasing resolutions. Subsequently, the path is determined on the low-resolution raster. By performing operations such as filtering directional points and mapping path points, the final path on the high-resolution raster can be obtained. The method enables a parallel computation of paths. Therefore, it significantly improves the efficiency for synthetic raster cost surfaces with continuous or discrete characteristics, as well as for raster cost surfaces generated from real terrain datasets, while also providing an end-to-end path output. The experiments show that 80% of the results are very close to the original LCP, and the accuracy of the remaining paths falls within an acceptable range. At the same time, our method greatly improves the efficiency of path solving on a large-scale raster, fulfilling practical application requirements.

**Keywords:** least-cost paths; raster cost surfaces; multi-resolution raster model; shortest-path problem



**Citation:** Tang, Q.; Dou, W. An Effective Method for Computing the Least-Cost Path Using a Multi-Resolution Raster Cost Surface Model. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 287. <https://doi.org/10.3390/ijgi12070287>

Academic Editors: Sisi Zlatanova and Wolfgang Kainz

Received: 23 May 2023

Revised: 8 July 2023

Accepted: 14 July 2023

Published: 17 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The least-cost path (LCP) is a common geographic analysis problem in raster-based geographic information systems (GIS) applied to infrastructure path planning, such as highways [1], power lines [2], and pipelines [3]. It is also used for wildlife conservation corridors [4] and exploring land routes in marine environments [5]. Additionally, different landscape paths can be defined by combining visual information with the least-cost path [6,7].

The least-cost path in raster space is a path consisting of adjacent cells from the starting point to the end point that have the least cumulative cost. Raster space divides a geographic area into equally sized squares, with each square capable of storing types of geographic features, such as elevation and land use. Connecting the center points of each square in the raster space generates a graph, with paths represented by a series of adjacent cells and implicitly hidden arc. The common connection structure between cells is the eight-neighborhood, which means that a cell is connected to its four orthogonal and four diagonal adjacent cells [8]. As for the situation where there are more movement directions between the cells, the 16-neighborhood can be applied. However, while applying a 16-neighborhood results in more accuracy in the path construction, the graph becomes more complex [9].

Raster cells are assigned a cost value or a risk indicator, and planners can determine the optimal spatial pathway through analyzing the distribution of these values. By evaluating the cost situation along the pathway, the quality of the route can be assessed [10]. Searching for the least-cost path on a cost surface can be seen as finding the shortest path on the graph. Hence, graph-based path search algorithms, such as Dijkstra's [11], A\* [12], D\* [13]

and theta\* [14] algorithms, can be applied. Graph-based path search algorithms commonly begin by checking if the starting node is the target node. If it is not, the search expands to neighboring nodes of the current node. Subsequently, a neighboring node is selected, and if it is not the target node, the search also expands to the neighboring nodes of this new node. The choice of nodes relies on various algorithms and their associated cost functions. This process is iterated until a path solution is found or the search process terminates after examining all nodes in the graph.

For graph-based path search algorithms, the environment is decomposed into cells, creating a defined configuration space. Changing the size of the configuration space incurs a time cost. Consequently, sample-based planning algorithms utilize random samples to represent the configuration space, which can solve various high-dimensional motion-planning problems [15,16]. The rapidly exploring random tree (RRT) algorithm [17,18] is employed to search for paths from a given starting point to a target location. The algorithm benefits from the strong exploratory capabilities of random points and utilizes a shortest-edge connectivity structure to simplify the analysis. The probabilistic road map (PRM) method [19] is employed for a path search between multiple starting points and multiple target points. It applies graph-based path search algorithms by randomly sampling and connecting points in free space prior to performing the path search. By integrating the random points with the obstacle areas, an abstracted path network graph is constructed to simplify the entire search space, making it more effective in solving paths with complex spatial configurations and constraints. Owing to the randomness of the sample points, variations in solutions for the same problem can arise, rendering this method unsuitable for stability-dependent scenarios.

Additionally, researchers take environmental factors into account when seeking the optimal path for specific application scenarios. For instance, Bagli et al. [2] assessed criteria related to human health, landscape, and ecosystems in order to plan power lines based on varying environments. Effat and Hassan [20] designed a railway route that minimized environmental impact by integrating model engineering, environmental, and hybrid perspectives, and eventually chose a hybrid route. Previously, the minimum cost path analysis primarily focused on terrestrial environments. Currently, maritime transportation has emerged as a prominent mode of transportation. Unlike land, the sea is characterized by a flat and slopeless terrain. This entails not only considering factors such as time, distance, energy consumption, and distance from land resources but also understanding the cultural aspects of maritime travel [21].

As raster accuracy improves, optimizing the algorithm's performance alone is insufficient to handle the computational burden of large-scale grids. The literature suggests that these improving LCP strategies can be divided into three groups: limiting the search range, decomposing the search problem and using "abstract" problem-solving strategies, as well as a combination of the above strategies [22,23]. Moreover, parallel processing using multiple hardware devices is also adopted as the main method [24,25].

Previous research has utilized "abstract" problem-solving strategies to create network abstractions through hierarchical structures and disregard irrelevant local details to decrease the graph's size, resulting in a reduced computation time. Likhachev and Ferguson [26] proposed an anytime-incremental search for autonomous vehicles on a multi-resolution lattice state space. Apart from using a high-resolution space near the robot or goal region, the search was conducted in a low-resolution space elsewhere. Botea et al. [27] proposed the hierarchical pathfinding A\* (HPA\*) method that incorporated the A\* algorithm and an abstraction strategy. The grid was partitioned into blocks of equal or irregular sizes [28] to build an abstract representation. Next, entrances along the block boundaries were identified and interconnected with the nodes on both the block and the internal boundaries. Subsequently, the initial and destination points were connected to the graph to create the final path. The method has undergone several improvements [29,30]. HPA\* is designed to perform path planning on a nonuniform cost grid and is primarily employed in the fields of robotics and gaming. Sánchez-Ibáñez et al. [31] proposed a novel path planning

method for planetary exploration rovers that utilizes a two-layer multi-resolution grid. MRA\* [32] reduces the computational cost of memory and preprocessing, and in the search process, it timely constructs a graph based on multi-resolution to ensure the integrity and suboptimality. Even though these methods all use the idea of multi-resolution hierarchical abstraction, our method is different in that it proposes a universal path computation framework that can be integrated with different raster-based path search algorithms. The specific process of the method also has its own characteristics, and parallel computing is introduced to greatly improve efficiency. The paper [33] proposed a method for path solving utilizing a multi-resolution raster model. However, it only incorporated the A\* search algorithm while considering slope and distance factors, without taking into account downsampling on the raster cost surface. Additionally, it diverged in terms of the criteria used for selecting the path's points. The experimental section in this paper is more extensive and detailed, with a comprehensive consideration of these factors.

This paper proposes an effective method for calculating the least-cost path using a multi-resolution raster cost surface model, which addresses the challenges encountered in large-scale raster computation. Through a series of computational experiments, we demonstrate the effectiveness of this approach. We refer to this method as multi-scale least-cost path (MS-LCP) since it computes the least-cost path on a multi-scale raster. The structure of this paper is organized as follows: In Section 2, we provide an overview of the fundamental theory of the least-cost path based on raster. Section 3 presents the detailed process of the proposed MS-LCP method, accompanied by the experimental process and results in Section 4. Section 5 discusses the advantages and limitations of our approach. Section 6 summarizes the primary findings of the experiments, identifies the shortcomings, and suggests future research directions.

## 2. Least-Cost Paths on Raster Cost Surfaces

A pixel is the smallest unit of measurement in a raster. The size of a pixel determines the level of information and resolution in a raster. Every pixel has a spatial resolution and attribute values. The spatial resolution pertains to the surface area that a single pixel covers. For instance, when the raster precision is set at 5 m, every pixel represents an area of 5 m squared. Increasing the spatial resolution improves the representation of surface features in a raster, at the cost of requiring more storage and processing power. The attribute values indicate the prominent features of the corresponding area, either in whole numbers or decimal points. As an illustration, each single pixel in a raster digital elevation model (DEM) corresponds to a square region. The elevation value within that region is the average or median.

A raster cost surface assigns different cost values to different locations in space. It is formed by combining raster landscape features in a weighted manner. The costs required for different landscape features are measured using various comparable units. Generally, the values of each landscape are scaled to the same range for weighting [34]. The cost value can represent the time and resource consumption required for the location in the path or space analysis. The problem of the shortest path on a grid-based map consists in finding the minimum length between two given points, where the map may include obstacles or diverse terrains. The algorithms commonly used to solve this task are Dijkstra's and A\* algorithms. To solve the least-cost path problem, adjusting the cost function of the nodes based on the fundamental principles of Dijkstra's algorithm is necessary. Assuming a path  $P$  is given on the cost surface  $f$ , the traditional least-cost path  $\text{Min } L_{\text{sum}}(P, f)$  is expressed as:

$$\text{Min } \sum_{i,j \in P} \left[ \left( \frac{f(i) + f(j)}{2} \right) \cdot l(i, j) \right] \quad (1)$$

Here,  $f(i)$  denotes the value of grid  $i$  on the cost surface  $f$ , and  $l(i, j)$  denotes the straight distance between grids  $i$  and  $j$ . The formula, known as "minimum-cost path" [10], is the LCP model that is compared with the method in this paper. Additionally, the optimality of

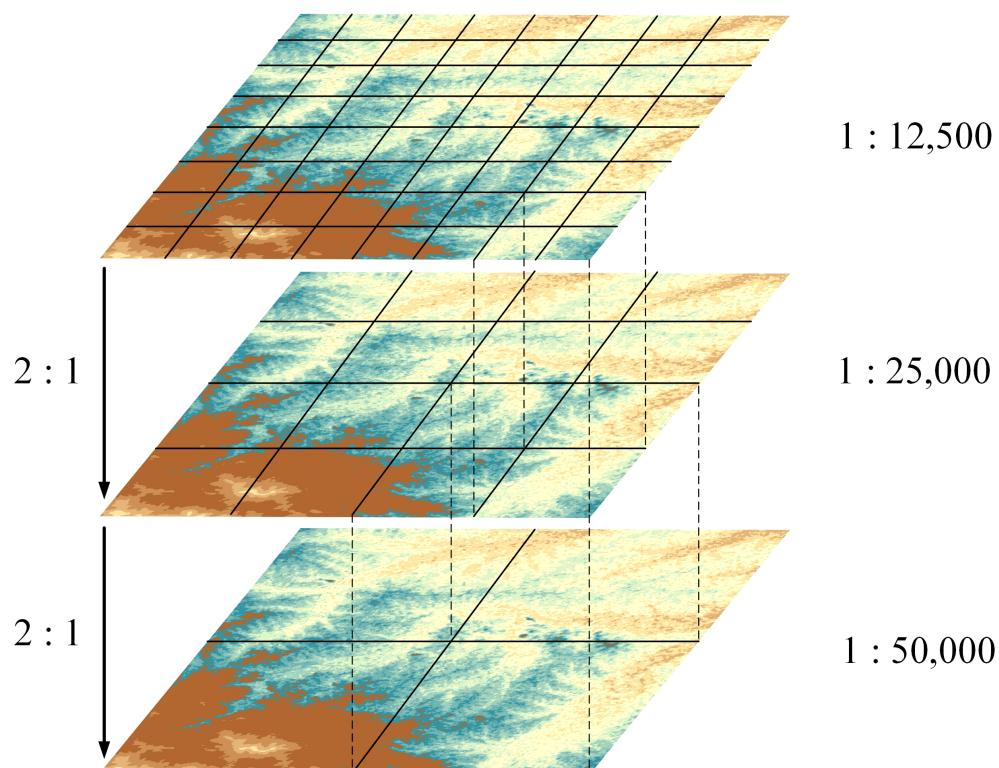
the minimum-cost path can be evaluated by minimizing the maximum cost length. This ensures that the cost of the selected path is not too high even in the worst-case scenario.

### 3. Method

#### 3.1. Overview of Proposed MS-LCP Method

The multi-scale least-cost path (MS-LCP) method utilizes a multi-resolution raster model for data processing prior to path calculation. The multi-resolution raster model gradually downsamples according to scale. The model is constructed by hierarchically processing the original high-resolution cost surface raster. In the downsampling process, multiple pixels from the high-resolution raster are compressed into one pixel in the low-resolution raster, thereby retaining rich information. The number of pixels in the raster progressively decreases, layer by layer.

When downsampling a raster, we first need to choose downsampling coefficients and methods. Common downsampling coefficients is 2:1, and methods such as the average and maximum can be used. Downsampling results in a change in scale. Selecting a 2:1 ratio factor, as shown in Figure 1, halves the number of pixels in the rows and columns, thereby reducing the original raster by half. Downsampling affects the spatial resolution as well. For instance, if a high-resolution raster has a 5-meter resolution, the spatial resolution becomes 10 m after using a 2:1 downsampling coefficient. Considering the required practical applications and the limitations of computational resources, it is essential to select appropriate downsampling methods and coefficients carefully.



**Figure 1.** A multi-resolution raster cost surface model (MRCSM) using a 2:1 downsampling coefficient.

The multi-resolution raster cost surface model (MRCSM) we constructed bears resemblance to a raster pyramid, yet there are distinctions between them. A raster pyramid model is frequently employed in geographic information systems (GIS) to improve the efficiency of the raster display. This is commonly accomplished through the utilization of three resampling techniques: nearest neighbor, bilinear, and bicubic convolution, wherein a 2:1 ratio is employed for successive downsampling. However, the MRCSM we devised

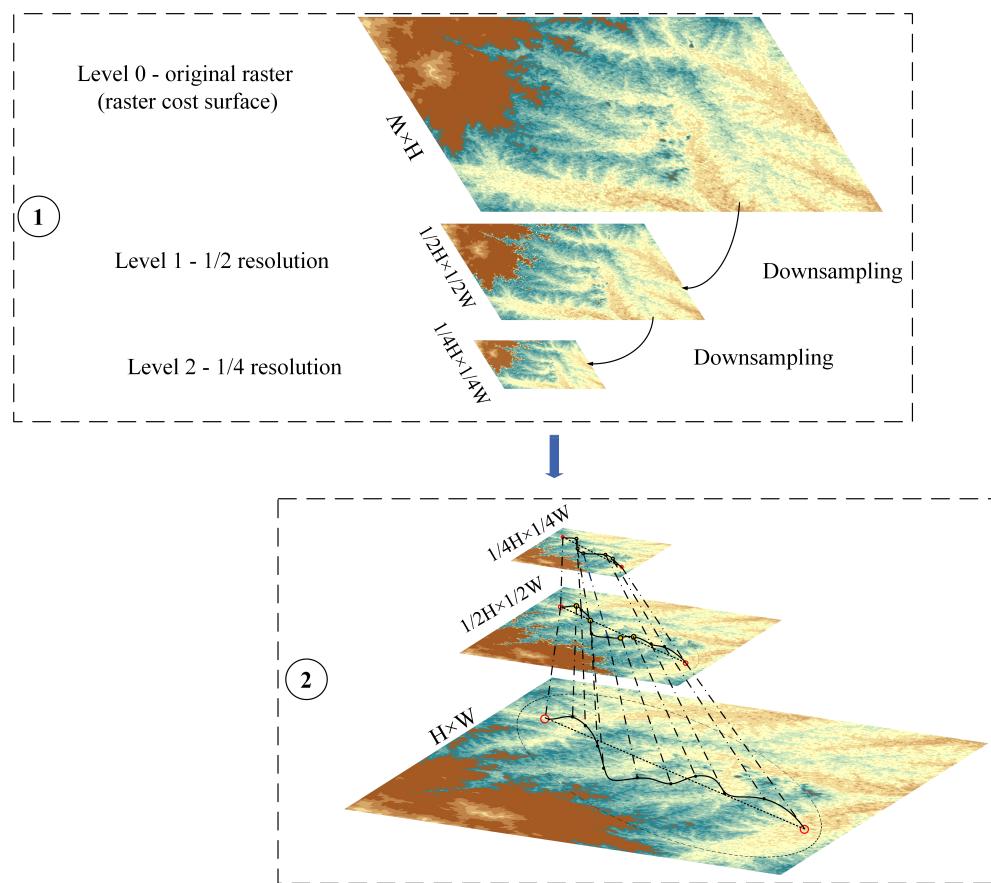
exclusively incorporates downsampling, wherein the use of average or maximum downsampling methods is permissible. Additionally, the downsampling ratio is not confined to a strict 2:1 proportion, affording flexibility for any adjustment.

Using a downsampling coefficient of 2:1 may result in the rows and columns of the original raster not being a multiple of 2. In such cases, the original raster is typically either padded or cropped to make it a multiple of 2, or the downsampling coefficient is directly changed (for example, to 3:1) to solve the problem. Cropping may have a serious impact on the subsequent path search and is not recommended. To aid in the mapping of path points between different-level rasters, we maintain a constant downsampling coefficient and perform padding on the original raster, copying the raster edge data from the last row or column value as well. It should be noted that values that are too large, too small, or special values (such as “no data”) should not be used for padding, otherwise the downsampling and path search will be affected, leading to edge effects on the path.

Excessive downsampling can result in the loss of important details in low-resolution grids. Therefore, in the proposed method, a threshold is set to control the level of the MRCMS. A low-resolution raster needs to be kept within a certain range in order to obtain an undistorted path. If the scale of the low-resolution raster is less than a certain value, better path results cannot be obtained, and this value can be the minimum scale threshold.

After constructing the MRCMS, the second stage involves a path calculation. The number of pixels directly affect the calculation and storage of the paths. Figure 2 ignores the spatial resolution of pixels and focuses on changes in the raster scale. We first solve the least-cost path at the lowest resolution to obtain a coarse-grained path. Then, in the following steps, we select some directional points on the coarse-grained path, and the collection of directional points represents the approximate direction of the path. After obtaining a series of directional points, it is mapped to the higher resolution raster of the next layer, and then the least-cost path between each pair of path points is calculated. The above steps are repeated until the final path of the original high-resolution raster has been obtained. The steps of the MS-LCP method are as follows:

- step 1. Acquired data from remote sensing are utilized to produce cost maps based on various criteria as input data.
- step 2. Set the threshold of the minimum resolution for the grid scale. Downsample the cost of each grid. Record the new scales of each layer grid and the positions of the starting and the target point in each layer. Use a grid-based path search algorithm.
- step 3. Traverse the grid layers from low to high. Calculate the path and record the sequence of paths starting from the minimum resolution grid.
- step 4. Filter the directional points and map the path points on the medium-high resolution grid one by one. Each time a new graph structure is established based on the path search range, calculate the paths between each group of path points serially or in parallel to obtain the final sequence of paths.
- step 5. Each layer can output the corresponding path map. Steps 2–4 complete the calculation of a cost map. In the case of multiple cost maps, iterate through steps 2–4.



**Figure 2.** The multi-scale least-cost path (MS-LCP) method: The first stage is to obtain a multi-resolution raster cost surface model of different scales using downsampling. The second stage is the path search on the raster cost surface at different scales.

### 3.2. Determination of Search Area

The raster-based least-cost path first needs to create a graph. Given the starting point and target point, we determine the search area of the path search algorithm to save computation time. The search area is usually a rectangle or polygon, where each unit or grid of raster is considered a path node, and all nodes form a shape. If the search area is a rectangle with the top-left corner node as  $(x_1, y_1)$  and the bottom-right corner node as  $(x_2, y_2)$ , then all the nodes it contains can be represented as:

$$S = \{(i, j) | x_1 \leq i \leq x_2, y_1 \leq j \leq y_2\} \quad (2)$$

Among them,  $S$  is the set of all nodes in the search area,  $i$  is the x-coordinate of the node, and  $j$  is the y-coordinate of the node. Therefore, the size of the search area can be obtained by calculating  $(x_2 - x_1 + 1) \times (y_2 - y_1 + 1)$ , which is the number of nodes.

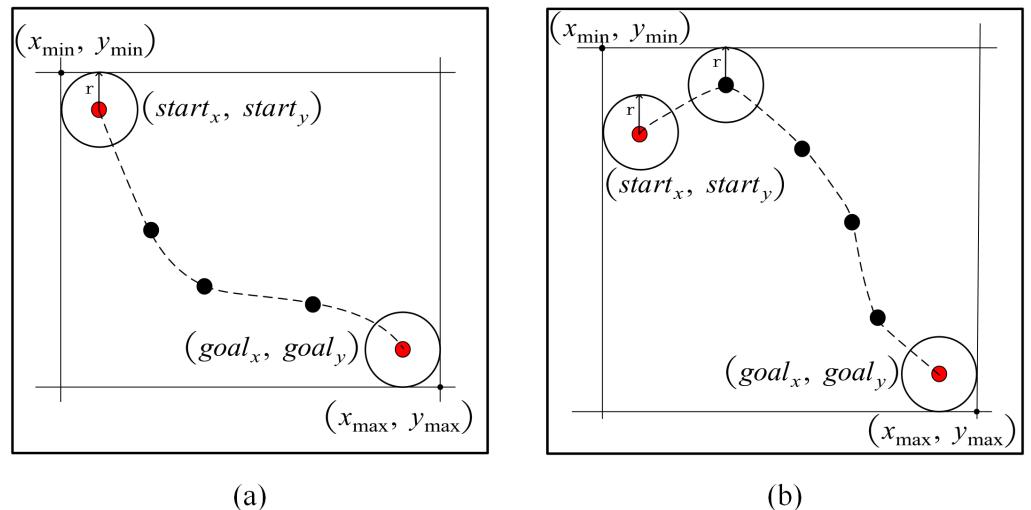
Assuming that the raster surface has a width of  $W$  and a height of  $H$ , each cell is of size  $s \times s$ . To determine the search area from starting point  $(start_x, start_y)$  to target point  $(goal_x, goal_y)$ , a rectangle is used. The search radius represented by  $r$  is the maximum distance that can be searched, defined as a multiple of  $s$  and calculated in grid units. A circle with a radius of  $r$  using the starting or target point as the center is drawn, and all cells encompassed by the circle are considered part of the search area, as shown in Figure 3a. The upper left cell of the rectangular area representing the search area is located at  $(x_{min}, y_{min})$ , and the lower right cell at  $(x_{max}, y_{max})$ . The upper left and lower right corner coordinates of the search area can be recalculated based on the starting point, target

point, and search radius  $r$ . The calculation method for the coordinates of the upper left cell is as follows:

$$\begin{aligned} x_{\min} &= \min(start_x, goal_x) - r/s \\ y_{\min} &= \min(start_y, goal_y) - r/s \\ \text{s.t. } x_{\min}, y_{\min} &\geq 0 \end{aligned} \quad (3)$$

The cell coordinates in the lower right corner are:

$$\begin{aligned} x_{\max} &= \max(start_x, goal_x) + r/s \\ y_{\max} &= \max(start_y, goal_y) + r/s \\ \text{s.t. } x_{\max} < H, y_{\max} < W \end{aligned} \quad (4)$$



**Figure 3.** Determination of the search area: (a) search area determined by the starting point and target point; (b) search area determined together by the starting point, target point, and other auxiliary edge points.

Therefore, the search area  $SA$  can be represented as:

$$SA = \{(i, j) | x_{\min} \leq i \leq x_{\max}, y_{\min} \leq j \leq y_{\max}\} \quad (5)$$

During the path calculation process on the MRCMSM, the search area on the lowest resolution raster can be determined as the area between the starting point and target point as mentioned above. However, the search area on subsequent higher resolutions may need to be further adjusted. As shown in Figure 3b, a point is needed to assist the above starting point ( $start_x, start_y$ ) and target point ( $goal_x, goal_y$ ) in the selected sequence of directional points for the path, in order to obtain the new search area formed by  $(x_{\min}, y_{\min})$  and  $(x_{\max}, y_{\max})$ .

### 3.3. Selection of Directional Points

Directional points differ from turning points on a path and the extraction method is relatively loose, including some not very obvious turning points. Consider a set of row and column coordinates on the path as  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ . Here,  $n$  represents the number of points along the path. This paper extracts directional points according to the following steps:

1. For each point on the path  $(x_i, y_i)$ , calculate its direction with the previous point  $(x_{i-1}, y_{i-1})$  using the vector representation  $\overrightarrow{P_i P_{i-1}} = (x_i - x_{i-1}, y_i - y_{i-1})$ .

2. For the points that are in the middle of the path  $(x_2, y_2), (x_3, y_3), \dots, (x_{n-1}, y_{n-1})$ , calculate the direction of the line connecting it with the previous point  $(x_{i-1}, y_{i-1})$  and the next point  $(x_{i+1}, y_{i+1})$ , that is,  $\overrightarrow{P_i P_{i-1}} = (x_i - x_{i-1}, y_i - y_{i-1})$  and  $\overrightarrow{P_{i+1} P_i} = (x_{i+1} - x_i, y_{i+1} - y_i)$ . Then, add these two vectors to obtain  $\overrightarrow{P_{i+1} P_{i-1}} = \overrightarrow{P_i P_{i-1}} + \overrightarrow{P_{i+1} P_i}$ . When the judgment condition  $\cos \theta = \frac{\overrightarrow{P_{i+1} P_{i-1}} \cdot \overrightarrow{P_i P_{i-1}}}{|\overrightarrow{P_{i+1} P_{i-1}}| |\overrightarrow{P_i P_{i-1}}|} \geq \cos 45^\circ = \frac{\sqrt{2}}{2}$ , and  $\cos \theta \neq 1$  is met, that is, the angle between  $\overrightarrow{P_{i+1} P_{i-1}}$  and  $\overrightarrow{P_i P_{i-1}}$  is no more than 45 degrees and greater than 0 degrees, the point is recorded as a path directional point.
3. The starting point and the target point should also be recorded as directional points, since they also represent the direction of the path.

The list of directional points obtained from the previous steps is collected. In addition to the method used to extract directional points from the path described above, polynomial curve fitting is an alternative method. This approach involves segmenting the path into line segments, performing a polynomial curve fitting on each segment, and filtering directional points after calculating the curvature. An appropriate curvature threshold must be chosen when using polynomial curve fitting to avoid issues resulting from exaggerated bending.

### 3.4. Mapping of Directional Points on Multi-Scale Raster

After obtaining the path nodes on the low-resolution grid, if the row and column values of the resulting path are different from the row and column values of the low-resolution grid, the row and column coordinates of the path nodes should be changed. For example, the row–column sequence of the search area determined on the low-resolution grid is  $(x_{\min}, y_{\min}), (x_{\min}, y_{\min+1}), \dots, (x_{\max}, y_{\max})$ , when a new graph structure is created from the search range and the path calculation is performed, if the rows and columns are redefined as starting from 0, the coordinates of the path directional points need to be magnified.

After obtaining the coordinates of appropriate path directional points, they are then mapped to a high-resolution raster. With a downsampling coefficient of 2:1 and using the averaging method, the row and column of a certain cell on the low-resolution raster are denoted as  $(x, y)$ , which correspond to  $(2x, 2y), (2x + 1, 2y), (2x, 2y + 1)$ , and  $(2x + 1, 2y + 1)$  on the high-resolution raster. If the path directional points are the starting and target point, they are mapped directly to the starting and target points of the high-resolution raster. For other directional points, the point with the minimum cost among the four located on the high-resolution raster is chosen.

### 3.5. Calculation Method between Multiple Sets of Path Points

To map a sequence of path directional points onto a high-resolution raster, computations are made between each pair of adjacent points along the path, and the resulting path values are connected to create the final path. The use of a serial calculation method for numerous sets of path points is inefficient. The parallelization of this task is deemed feasible if it can be decomposed into independent, noncommunicating subtasks. The results of the subtasks for each path must merge seamlessly without any data-dependent conflicts.

We divide the multiple path calculation tasks into left and right subtrees, recursively compute the least-cost paths of each, and then merge the results. We set a threshold based on the number of selected directional points to determine whether to parallelize the path calculation. If the number of directional points is less than 4, parallel resources are unnecessary. By adjusting the threshold, the path calculation can be configured to be either serial or parallel. Further optimizations can be achieved by fine-tuning the task partitioning and load balancing to take full advantage of the strengths of parallel computing technology.

Simultaneously during multi-threaded parallel computing, the search area may affect the process. Two scenarios exist in our computation processes. In one scenario, each thread acquires the search area of the complete high-resolution raster, while in the other scenario, each thread determines the search area based on individual path-point groups. We conducted experiments on both scenarios. The first scenario offers a higher path

accuracy than the latter, while the latter provides a faster efficiency. Balancing efficiency and accuracy, our method, in the following section, employs the initial scenario for testing. We simultaneously employed Dijkstra's algorithm as the path search algorithm in our experiment. Furthermore, we can employ different path search algorithms, such as the A\* algorithm and dynamic programming algorithm, without jeopardizing the efficiency and functionality of our method.

#### 4. Experiments

To validate the effectiveness of our approach, three computational experiments were conducted, utilizing Dijkstra's algorithm as the path search in all of them. We selected the widely employed downsampling methods of averaging, maximum values, and a 2:1 proportion factor for downsampling the raster cost surface. The initial experiment entailed a comparison of the path accuracy using synthetic data. The subsequent experiment examined the efficiency of the method. The final experiment aimed to assess the limits of the proposed method using real data. Java was employed as the programming language for all experiments, conducted on a computing system featuring an Intel Core i7-12700H CPU processor operating at 2.30 GHz and equipped with 32 GB of memory.

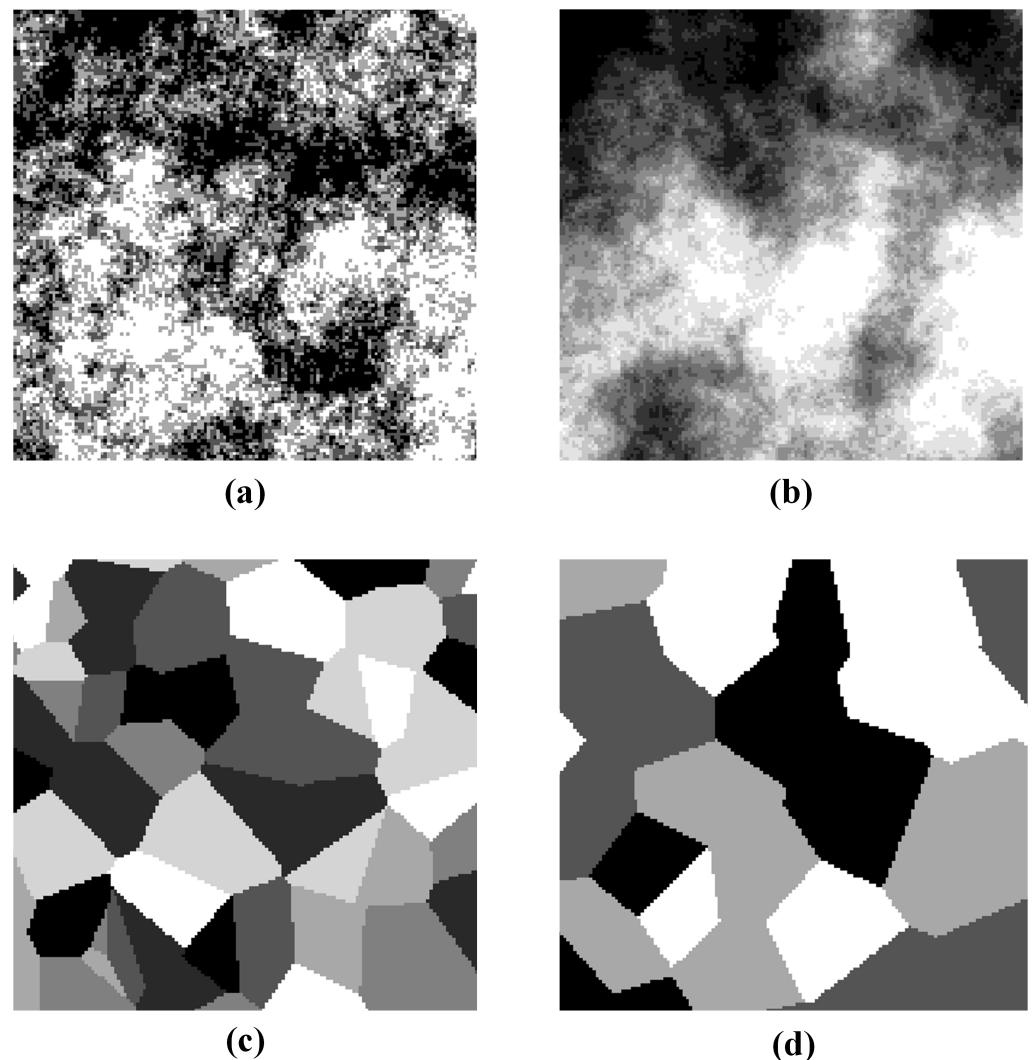
##### 4.1. Data

###### 4.1.1. Experiment 1

Realistic neutral landscape models with diverse spatial configurations can be generated by employing multiple algorithms. As described in the work of Murekatete and Shirabe [10], we generated two categories of random raster cost surfaces, totaling one thousand, using the midpoint displacement method [35] and the random-cluster nearest-neighbor algorithm provided by PYTHON software package, NLMpy [36]. We discretized the random values into equidistant integer intervals and saved them as raster files. These raster surfaces exhibited two distinct spatial structures: one with a continuous gradient of values (named "cloudy") and the other with constant values separated by clear boundaries (named "patchy").

The "cloudy" raster was generated using the mid-point displacement method provided by the NLMpy function. The function takes three parameters and generates a grid with randomly assigned values ranging from zero to one. The parameters nRow and nCol determine the dimensions of the grid, and the parameter h controls the level of spatial autocorrelation. Moreover, the "patchy" raster was created by utilizing the random-cluster nearest-neighbor algorithm, which partitions the grid into irregular shapes of relatively consistent dimensions. The parameter s was used to control the approximate size of the patch. After obtaining the grids within the specified range, both types were converted into cost surfaces. The cost values for the two types of cost surfaces ranged from 1 to 100, and the class of values was determined by the number of unique values in the synthetic landscape raster.

This experiment involved generating 1000 sets of cost surface data for the two types separately. Moreover, the number of rows (nRow) and columns (nCol) both equaled 200. Figure 4 showcases examples of cost surfaces with various classes, including "cloudy" and "patchy" types.



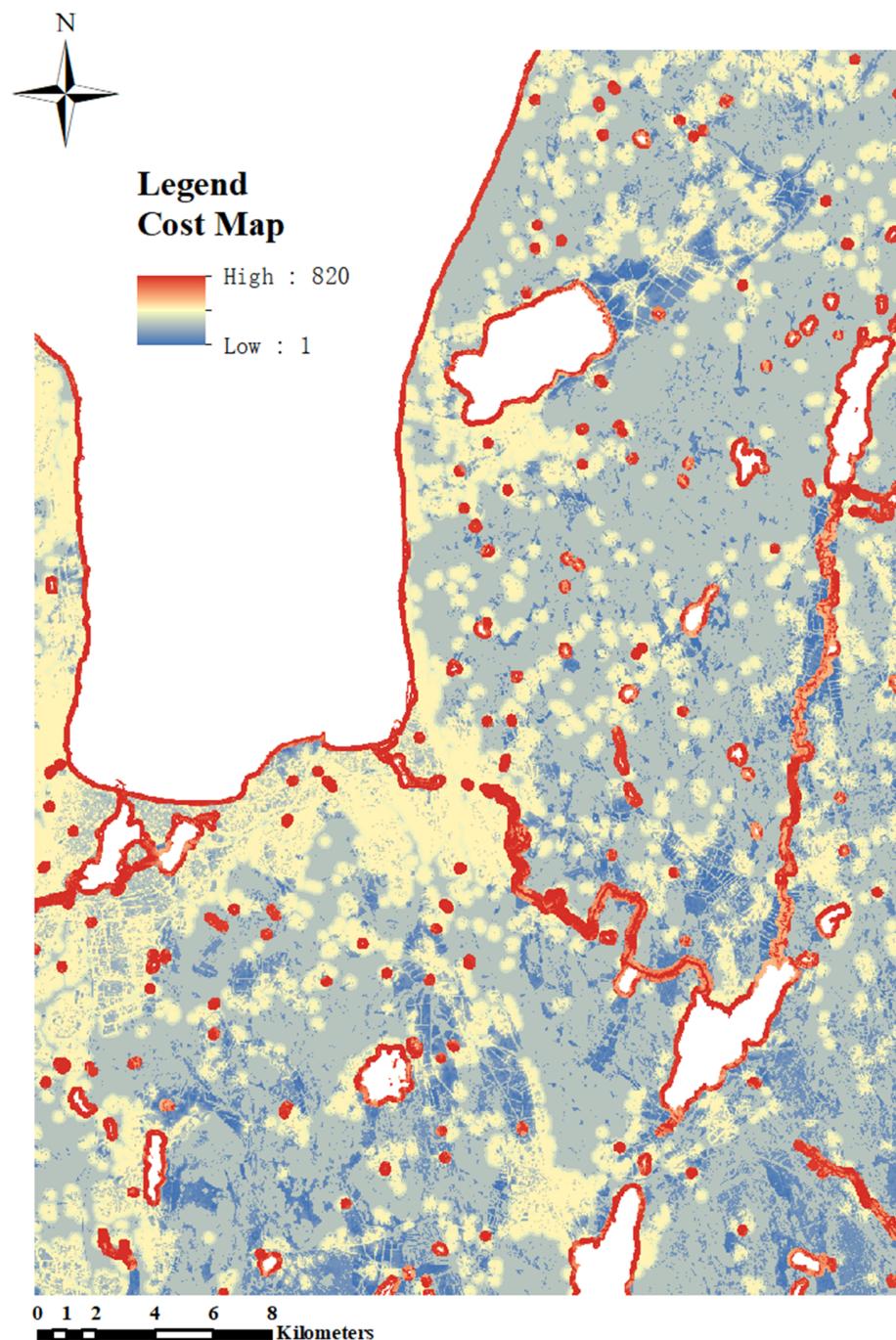
**Figure 4.** Examples of (a,b) cloudy cost surfaces and (c,d) patchy cost surfaces: (a) 4 classes, (b) 10 classes, (c) 7 classes, and (d) 4 classes. Darker shades represent higher values.

#### 4.1.2. Experiment 2

We created three additional sets of data, each containing 200 cloudy cost surfaces, with cell sizes of  $250 \times 250$ ,  $500 \times 500$ , and  $750 \times 750$ , respectively. The experiment was run on a computer with 14 cores and 20 threads, as well as 32 GB of memory.

#### 4.1.3. Experiment 3

A cost surface comprising three datasets, elevation, land cover, and infrastructure, was obtained for road planning, aiming to assess the performance of the proposed method on actual data [37]. The cost surface could be weighted based on various criteria, including water, land cover, slope, and buildings. For this experiment, a cost surface of around 500 square kilometers was utilized, featuring a spatial resolution of 20 m and cost values ranging from 1 to 820. Figure 5 shows that water sources contribute the highest proportion of the cost in this cost surface, followed by land use.



**Figure 5.** Raster cost surface comprising three datasets: elevation, land cover, and infrastructure.

#### 4.2. Results

##### 4.2.1. Experiment 1

On the cloudy and patchy cost surfaces, we obtained 1000 pairs of LCP and serial MS-LCP path results and calculated the total cost length of each path. Meanwhile, we adopted the approach proposed by Murekatete and Shirabe [10] to measure the maximum cost length and measured the segment length that intersected with the “undesirable” cell, which is greater than or equal to the maximum cost length of the LCP, for comparison.

Tables 1 and 2 show the partial path results for 1000 pairs of LCPs and MS-LCPs generated on cloudy and patchy cost surfaces. Among them, the MS-LCP selected two types of downsampling methods: average and maximum. The four groups of data chose the same set of starting and target points to compare the path under different downsampling

methods.  $l(LCP)$  and  $l(MS-LCP)$  represent the cost length of the total path, and the ratio of two paths' total lengths is shown as  $l(MS-LCP)/l(LCP)$ .  $u(LCP)$  and  $u(MS-LCP)$  represent the length of fragments intersecting with “undesirable” cells. Similarly, the ratio between them is shown as  $u(MS-LCP)/u(LCP)$ .

**Table 1.** Results on cloudy or patchy cost surfaces and the multi-scale least-cost path (MS-LCP) using the average downsampling method.

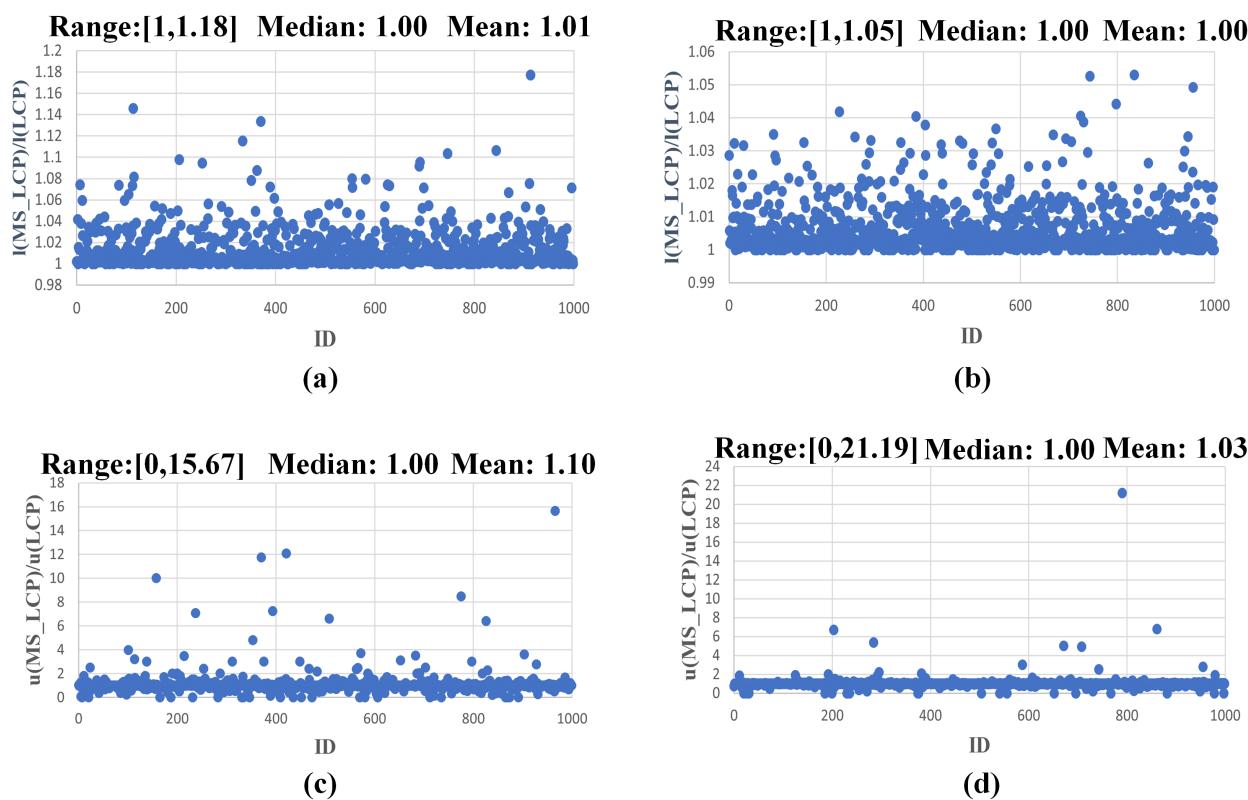
Cost Surfaces	ID	$l(LCP)$	$l(MS-LCP)$	$u(LCP)$	$u(MS-LCP)$
Cloudy	0	10,565.57	10,585.46	32.06	33.51
	1	17,893.35	17,925.78	61.10	59.69
	2	10,890.05	11,343.96	13.31	13.07
	3	17,703.28	17,703.28	29.70	29.70
	...	...	...	...	...
	996	2918.50	3126.06	20.09	24.68
	997	8220.12	8222.81	45.13	45.34
	998	17,296.15	17,351.81	48.38	48.96
	999	15,812.72	15,812.72	29.70	29.70
Patchy	0	6376.10	6558.65	8.50	6.12
	1	8236.46	8284.00	17.26	17.26
	2	11,515.55	11,539.67	53.16	50.75
	3	10,448.90	10,505.47	9.71	9.71
	...	...	...	...	...
	996	14,437.07	14,470.38	37.36	36.53
	997	9248.01	9423.55	2.83	0.00
	998	9165.71	9249.36	8.95	9.36
	999	15,903.55	15,903.55	55.33	55.33

**Table 2.** Results on cloudy or patchy cost surfaces and the multi-scale least-cost path (MS-LCP) using the maximum downsampling method.

Cost Surfaces	ID	$l(LCP)$	$l(MS-LCP)$	$u(LCP)$	$u(MS-LCP)$
Cloudy	0	10,565.57	10,738.45	32.06	40.96
	1	17,893.35	17,930.61	61.10	59.69
	2	10,890.05	11,450.37	13.31	14.69
	3	17,703.28	17,703.28	29.70	29.70
	...	...	...	...	...
	996	2918.50	3311.17	20.09	25.68
	997	8220.12	8261.94	45.13	47.13
	998	17,296.16	17,352.01	48.38	49.17
	999	15,812.72	15,812.72	29.70	29.70
Patchy	0	6376.10	6589.71	8.50	6.12
	1	8236.46	8329.78	17.26	17.26
	2	11,515.55	11,550.11	53.16	53.16
	3	10,448.90	10,558.63	9.71	9.71
	...	...	...	...	...
	996	14,437.07	14,470.38	37.36	36.53
	997	9248.01	9381.04	2.83	0.00
	998	9165.71	9225.55	8.95	10.19
	999	15,903.55	15,903.55	55.33	55.33

Figures 6 and 7 illustrate visualization results of 1000 data points in the dimensions of  $l(MS-LCP)/l(LCP)$  and  $u(MS-LCP)/u(LCP)$ , respectively. The corresponding ranges, medians, and averages are labeled. It should be noted that for each set of data, the total length of the path in 80% of the MS-LCP results is only slightly different from that of the LCP. The averages and medians are both close to one, and the value range is [1, 1.33]. The  $u(MS-LCP)/u(LCP)$  dimension has value ranges of [0, 15.67], [0, 21.19], [0, 30.83], and [0, 21.90]. Additionally, there are cases where the length of the intersection between

the maximum downsampled MS-LCP and the “undesirable” cell exceeds the average downsampled MS-LCP.

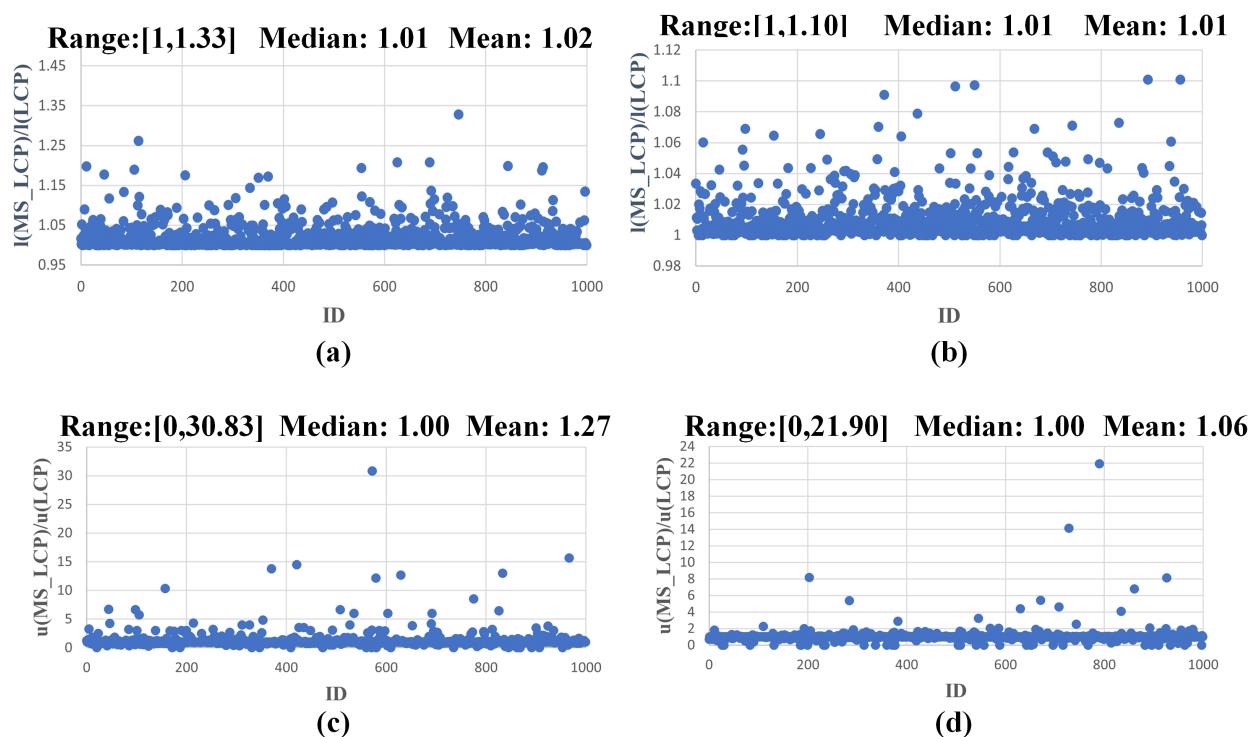


**Figure 6.** The y-axis shows  $l(\text{MS} - \text{LCP})/l(\text{LCP})$  or  $u(\text{MS} - \text{LCP})/u(\text{LCP})$ , and the x-axis shows the ID. The plotted chart displays the range, median and mean values. (a,c): results on cloudy cost surfaces and the MS-LCP using average downsampling; (b,d): results on patchy cost surfaces and the MS-LCP using average downsampling.

In addition, we compared the maximum value, minimum value, and Pth percentile of path cost values to comprehensively compare the cost value composition of different paths. In Tables 3 and 4, we compared the maximum and minimum cost values of two different paths, as well as the 75th, 50th, and 25th percentiles. Based on the cloudy and patchy cost surfaces, we considered the impact of different downsampling methods.

**Table 3.** Results on cloudy or patchy cost surfaces and MS-LCP using the average downsampling method.

Cost Surfaces	Values	LCP < MS-LCP	LCP = MS-LCP	LCP > MS-LCP
Cloudy	Maximum	28	950	22
	75th percentile	47	911	42
	50th percentile	38	915	47
	25th percentile	21	951	28
	Minimum	4	987	9
Patchy	Maximum	17	965	18
	75th percentile	29	947	24
	50th percentile	17	961	22
	25th percentile	15	970	15
	Minimum	2	996	2



**Figure 7.** The y-axis shows  $l(MS - LCP)/l(LCP)$  or  $u(MS - LCP)/u(LCP)$ , and the x-axis shows the ID. The plotted chart displays the range, median and mean values. (a,c): results on cloudy cost surfaces and MS-LCP using maximum downsampling; (b,d): result on patchy cost surfaces and MS-LCP using maximum downsampling.

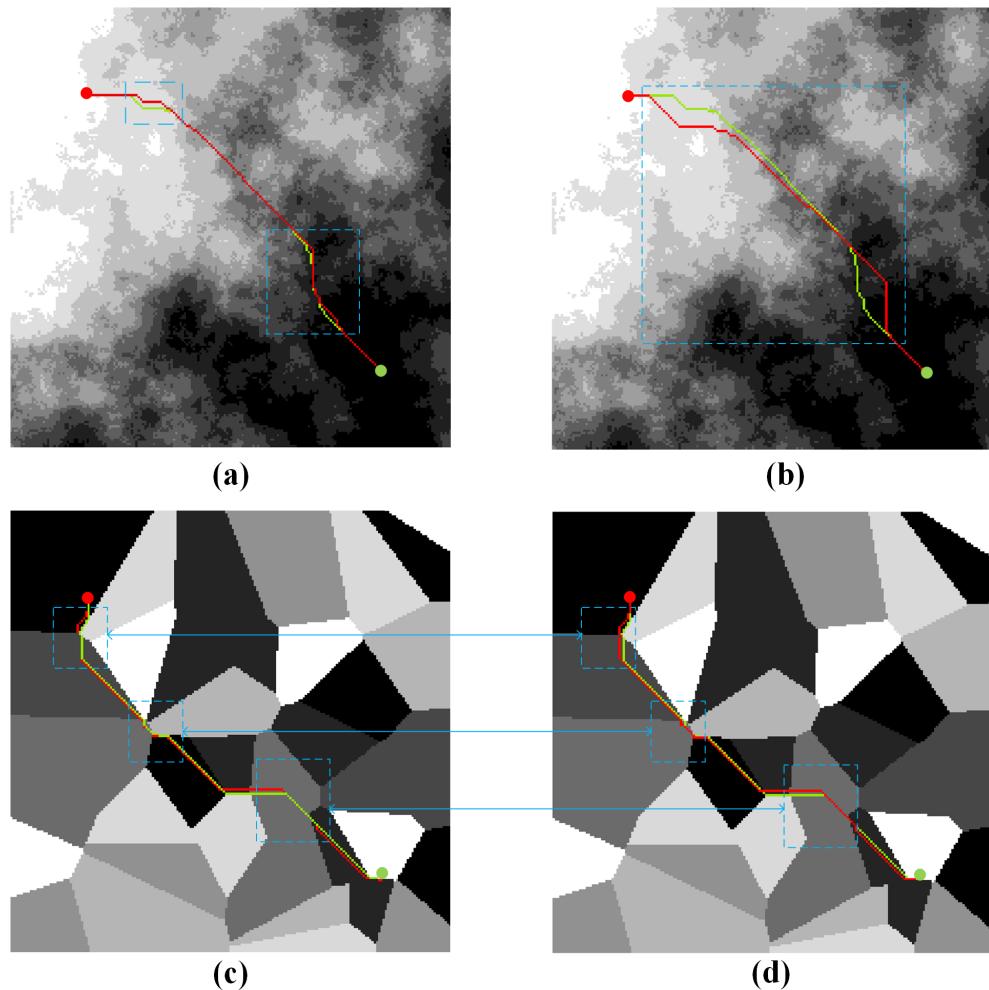
In Table 3, when the cost surface is “cloudy”, there are 28 cases where the maximum cost value of the LCP is lower than that of the MS-LCP; there are 950 cases where the maximum cost value of the LCP is equal to that of the MS-LCP; and in the remaining 22 cases, the maximum cost value of the LCP is higher than that of the MS-LCP. Other data in the table can be explained similarly. From the table, it can be seen that the difference in path cost between the LCP and the MS-LCP on the patchy cost surface is smaller than the difference on the cloudy cost surface. The path accuracy of MS-CP based on the average downsampling is higher than that of the maximum downsampling.

**Table 4.** Results on cloudy or patchy cost surfaces and MS-LCP using the maximum downsampling method.

Cost Surfaces	Values	LCP < MS-LCP	LCP = MS-LCP	LCP > MS-LCP
Cloudy	Maximum	47	935	18
	75th percentile	71	884	45
	50th percentile	48	898	54
	25th percentile	35	934	31
	Minimum	4	987	9
Patchy	Maximum	18	951	31
	75th percentile	47	912	41
	50th percentile	35	935	30
	25th percentile	25	958	17
	Minimum	6	993	1

Figure 8 illustrates four path scenarios with identical starting and target points. The green path represents the LCP, while the red path represents the MS-LCP. The blue dotted box indicates the distinctions between the two paths. On the cloudy cost surface,

the MS-LCP is better suited for the average downsampling. On the patchy cost surface, the various downsampling techniques of the MS-LCP method yield favorable outcomes.



**Figure 8.** (a) LCP and MS-LCP using average downsampling on cloudy cost surface. (b) LCP and MS-LCP using maximum downsampling on cloudy cost surface. (c) LCP and MS-LCP using average downsampling on patchy cost surface. (d) LCP and MS-LCP using maximum downsampling on patchy cost surface.

#### 4.2.2. Experiment 2

On the raster cost surface with  $250 \times 250$ ,  $500 \times 500$ , and  $750 \times 750$  scales, two different random cells were selected to calculate the LCP and parallel MS-LCP on the same computing platform of the first experiment. Since there was little difference in time between average and maximum downsampling, we compared the average calculation time of the LCP and average downsampled MS-LCP, as shown in Table 5. The time of the MS-LCP includes reading the raster, downsampling, determining the search area, selecting directional points and mapping, as well as constructing graphs and calculating the path. The time of the LCP includes reading the raster, constructing graphs, and calculating the path.

Table 5 reveals that the average calculation time for the LCP increased gradually as the raster scale increased; conversely, the average calculation time for the MS-LCP increased only marginally. The LCP's calculation time for each raster was relatively consistent, while the MS-LCP's calculation time varied significantly. The LCP relied on the complete diagram of the raster construction process to perform calculations, whereas the MS-LCP set the

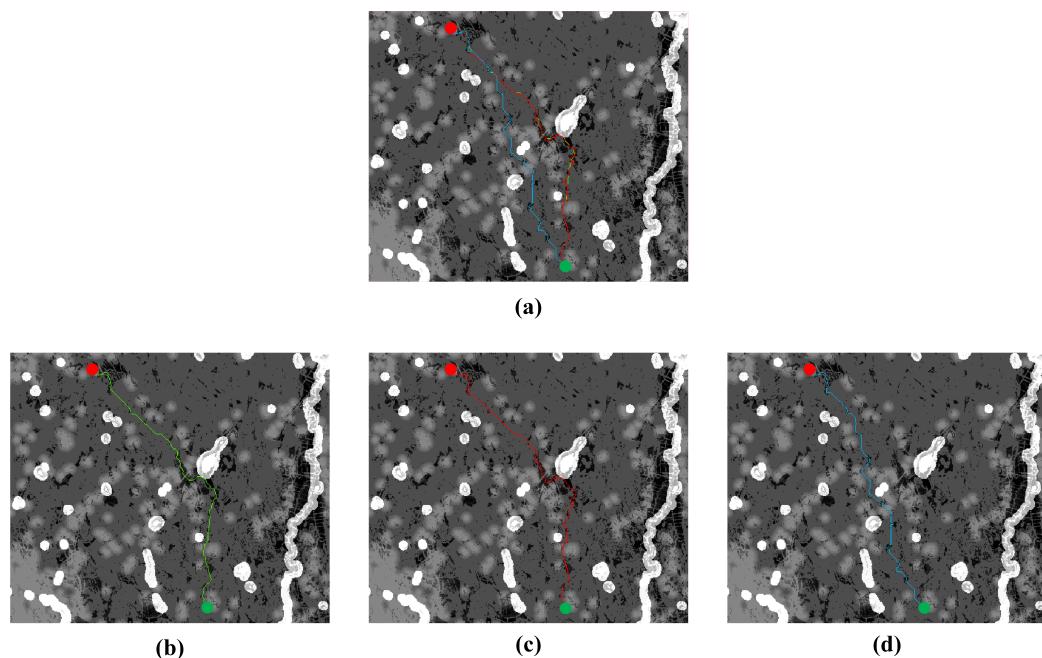
search area and parallelizable characteristics, making it better-suited for large-scale data processing under optimal computing power and storage conditions.

**Table 5.** Results on cloudy cost surfaces and MS-LCP using the average downsampling method.

Methods	Rows and Columns of Cost Surfaces		
	250 × 250	500 × 500	750 × 750
LCP	0.25 s	4.06 s	20.92 s
MS-LCP	0.07 s	0.09 s	0.26 s

#### 4.2.3. Experiment 3

Path search results were obtained for both the traditional LCP method and the proposed MS-LCP method using real terrain data. Different starting and target points were randomly selected for each set. Tables 6 and 7 display the ratios of  $I(\text{MS-LCP})/I(\text{LCP})$  and  $u(\text{MS-LCP})/u(\text{LCP})$  for the 1000 datasets under each downsampling methods, along with the corresponding ranges, medians, and means. Figure 9 illustrates the paths that correspond to the results mentioned above.



**Figure 9.** (a) Mixture of three different paths. (b) LCP on raster cost surface. (c) MS-LCP using average downsampling on raster cost surface. (d) MS-LCP using maximum downsampling on raster cost surface.

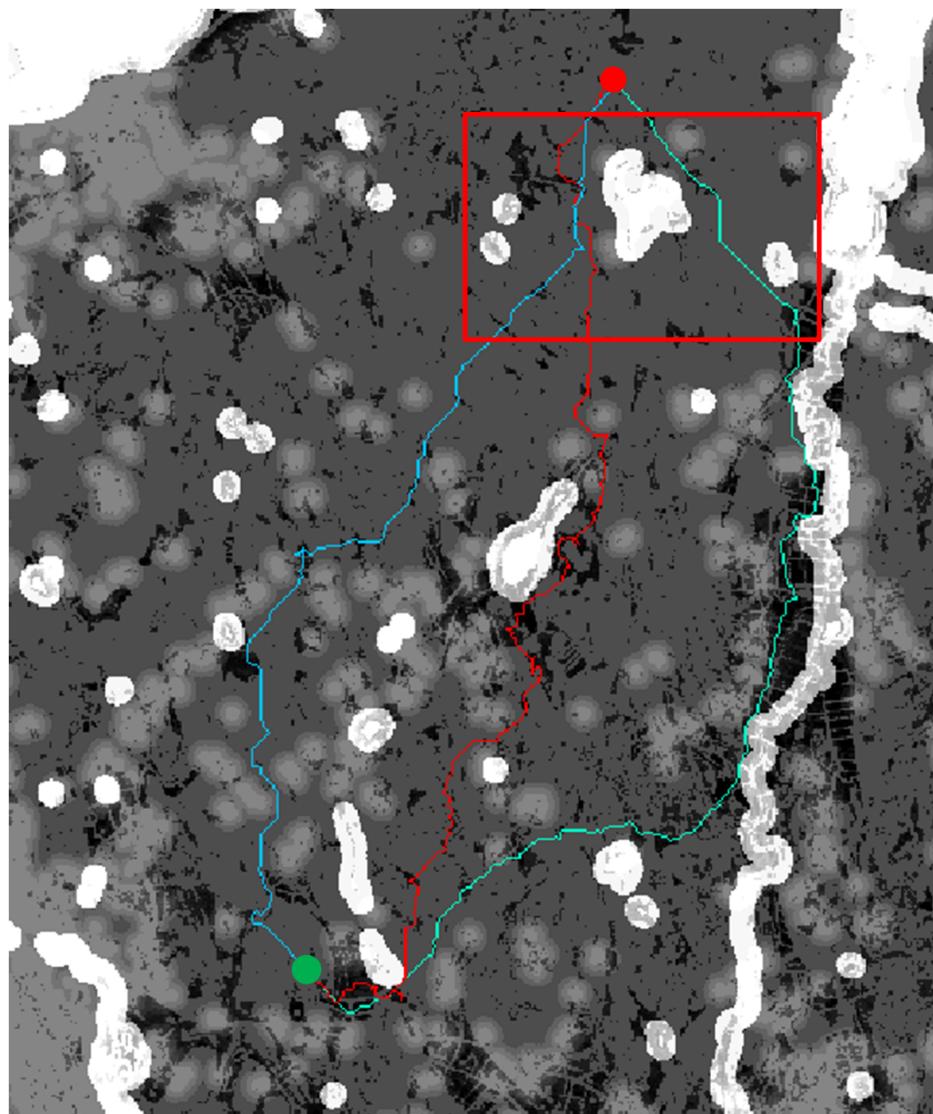
**Table 6.** LCP and MS-LCP using the average downsampling method.

	Range	Median	Mean
$I(\text{MS-LCP})/I(\text{LCP})$	(1, 1.60)	1.04	1.05
$u(\text{MS-LCP})/u(\text{LCP})$	(0, 6.24)	0	0.42

**Table 7.** LCP and MS-LCP using the maximum downsampling method.

	Range	Median	Mean
$I(\text{MS-LCP})/I(\text{LCP})$	(1, 1.53)	1.08	1.09
$u(\text{MS-LCP})/u(\text{LCP})$	(0, 11)	0	0.41

Based on the data in the table, it is evident that for actual terrain data, the majority of path search results obtained using the MS-LCP method fell within an acceptable range. When the search range between the starting point and the target point became smaller (i.e., the starting point was in close proximity to the target point, indicating a narrower data range), the total cost length of MS-LCP increased significantly compared to that of the LCP. Due to the downsampling being applied before each path calculation, the MS-LCP was not suitable for datasets with a small range. Furthermore, during the experiment, it was observed that due to the presence of obstacles on the cost surface (which signified impassable areas), the average downsampled MS-LCP with good performance may exhibit substantial discrepancies in overall path direction, when compared to the LCP, as depicted in the Figure 10. However, the total cost length of the MS-LCP at that point did not exhibit substantial differences compared to that of the LCP. Owing to the continuous downsampling, the extent of obstacle areas gradually diminished, and even the grid with the lowest resolution may contain no obstacle areas. Consequently, the MS-LCP progressively deviated from the intended direction during point mapping across multiple resolution grid layers.



**Figure 10.** The green path represents the LCP, the red path represents the MS-LCP using an average downsampling, and the blue path represents the MS-LCP using a maximum downsampling. The red rectangular area indicates the initial divergence point of the three paths. Within the cost grid, the white areas signify areas obstructed by water.

## 5. Discussion

On the patchy cost surface, both the average and maximum downsampled MS-LCP paths achieved good results. On the cloudy cost surface, the MS-LCP obtained by the average downsampling was more accurate and reliable than using the maximum downsampling. This could be attributed to the fact that the average downsampling smooths the details in the raster to a certain extent and preserves periodic fluctuations or noise. However, the maximum downsampling ignores some noise and detail information by only using the maximum value, introducing discontinuities and reducing the accuracy of the path calculation.

The continuous downsampling of the raster can introduce errors that divert the path from the optimal solution, leading to a reduction in accuracy. While downsampling improves efficiency, it is important to strike a balance to avoid blindly prioritizing efficiency over accuracy. On the low-resolution grid, there are no directional points along the path except for the start and end points, which indicates that the path is closer to a straight line. In this situation, the additional downsampling process will decrease the efficiency of the method. Conversely, if more path directional points are selected, suggesting multiple bends in the path, the efficiency of the path will also be reduced. These two scenarios represent extreme cases. When there is sufficient computational power and storage capacity, efficiency can still be improved. However, if the selection of path directional points is too dense, the final path will be more curved, necessitating path smoothing. Experimental results revealed that obstacle regions in the cost grid could influence the overall direction of the path. This occurred because as the downsampling of the obstacle region proceeded, the region gradually shrunk. Although there was a significant change in the overall direction of the path, the total cost length of the path did not deviate significantly.

Building a multi-resolution raster cost surface model consists in preprocessing the abstract representation of maps at various levels. It heavily relies on smooth transitions between high- and low-resolution abstractions, requiring a significant amount of memory to maintain the abstractions of each layer and the associated computational cost of preprocessing. However, our method aims to enhance the efficiency of path searches in large-scale maps. By utilizing a multi-resolution raster model, determining the search area, and parallelizing path computations, the efficiency is significantly improved. Many scholars have proposed schemes for multi-resolution or variable-resolution representations of maps. However, their computational efficiency with large-scale data is not comparable because our method can rapidly achieve efficiency. In addition to parallel computing in path solving, it can also be utilized for path point mapping.

The analysis of the least-visible path (LVP) aims to determine the path with the lowest cost on a visibility-associated cost surface [7]. In the traditional least-visible path approach, the cost surface is calculated by obtaining visibility information for all terrain points in a raster digital elevation model (DEM). The viewshed matrix utilizes the line of sight to compute the visibility between grid cells. Our proposed method can provide a solution for determining the least-cost path on the large-scale viewshed map.

## 6. Conclusions

The proposed multi-scale least-cost path (MS-LCP) method aims to perform a path search on large-scale cost surface and is a general computational framework. Firstly, the cost raster is preprocessed using different downsampling methods. A rough path is obtained on a low-resolution raster representing coarse-grained information. By defining the search area, filtering path directional points, and mapping path points, the final path on a high-resolution raster is determined. Experimental results on synthetic and real data demonstrated the effectiveness of the method. The efficiency of the method on large-scale data can be significantly improved through parallel computing.

Overall, the MS-LCP method is very versatile for least-cost path calculations, considering multiple factors and producing an end-to-end path output. We mainly conducted experiments using a downsampling ratio of 2:1, but other ratios can be attempted. In the

directional point filtering process, we did not consider the density of the path points, which can be further optimized. This study is applicable to the fields of geography and GIS. It requires specific storage conditions and computing resources for effective path planning over large areas. Furthermore, this study can be expanded to serve broader purposes.

**Author Contributions:** Formal analysis, Qiuling Tang; methodology, Wanfeng Dou; software, Qiuling Tang; writing—original draft, Qiuling Tang; writing—review and editing, Wanfeng Dou. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (No. 41930102, No. 41771411).

**Data Availability Statement:** All data are available online. The respective origins of these data are stated in this article.

**Acknowledgments:** The authors are grateful to the reviewers for their insightful comments, which significantly improved the quality of this paper.

**Conflicts of Interest:** No potential conflict of interest was reported by the authors.

## References

- Scaparra, M.P.; Church, R.L.; Medrano, F.A. Corridor location: The multi-gateway shortest path model. *J. Geogr. Syst.* **2014**, *16*, 287–309. [[CrossRef](#)]
- Bagli, S.; Geneletti, D.; Orsi, F. Routeing of power lines through least-cost path analysis and multicriteria evaluation to minimise environmental impacts. *Environ. Impact Assess. Rev.* **2011**, *31*, 234–239. [[CrossRef](#)]
- Durmaz, A.I.; Ünal, E.Ö.; Aydin, C.C. Automatic pipeline route design with multi-criteria evaluation based on least-cost path analysis and line-based cartographic simplification: A case study of the mus project in Turkey. *Isprs Int. J.-Geo-Inf.* **2019**, *8*, 173. [[CrossRef](#)]
- Chetkiewicz, C.L.B.; Boyce, M.S. Use of resource selection functions to identify conservation corridors. *J. Appl. Ecol.* **2009**, *46*, 1036–1047. [[CrossRef](#)]
- Gustas, R.; Supernant, K. Least cost path analysis of early maritime movement on the Pacific Northwest Coast. *J. Archaeol. Sci.* **2017**, *78*, 40–56. [[CrossRef](#)]
- Stucky, J.L.D. On applying viewshed analysis for determining least-cost paths on digital elevation models. *Int. J. Geogr. Inf. Sci.* **1998**, *12*, 891–905. [[CrossRef](#)]
- Lu, M.; Zhang, J.; Lv, P.; Fan, Z. Least visible path analysis in raster terrain. *Int. J. Geogr. Inf. Sci.* **2008**, *22*, 645–656. [[CrossRef](#)]
- Xu, J.; Lathrop, R.G., Jr. Improving simulation accuracy of spread phenomena in a raster-based geographic information system. *Int. J. Geogr. Inf. Syst.* **1995**, *9*, 153–168. [[CrossRef](#)]
- Seegmiller, L.; Shirabe, T.; Tomlin, C.D. A method for finding least-cost corridors with reduced distortion in raster space. *Int. J. Geogr. Inf. Sci.* **2021**, *35*, 1570–1591. [[CrossRef](#)]
- Murekatete, R.M.; Shirabe, T. An experimental analysis of least-cost path models on ordinal-scaled raster surfaces. *Int. J. Geogr. Inf. Sci.* **2021**, *35*, 1545–1569. [[CrossRef](#)]
- Dijkstra, E. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [[CrossRef](#)]
- Kala, R.; Shukla, A.; Tiwari, R. Fusion of probabilistic A\* algorithm and fuzzy inference system for robotic path planning. *Artif. Intell. Rev.* **2010**, *33*, 307–327. [[CrossRef](#)]
- Stentz, A. Optimal and efficient path planning for partially-known environments. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; pp. 3310–3317.
- Daniel, K.; Nash, A.; Koenig, S.; Felner, A. Theta\*: Any-angle path planning on grids. *J. Artif. Intell. Res.* **2010**, *39*, 533–579. [[CrossRef](#)]
- Elbanhawi, M.; Simic, M. Sampling-based robot motion planning: A review. *IEEE Access* **2014**, *2*, 56–77. [[CrossRef](#)]
- Kingston, Z.; Moll, M.; Kavraki, L.E. Sampling-based methods for motion planning with constraints. *Annu. Rev. Control. Robot. Auton. Syst.* **2018**, *1*, 159–185. [[CrossRef](#)]
- LaValle, S. Rapidly-Exploring Random Trees: A new Tool for Path Planning. *Research Report 9811*. 1998. Available online: <https://cir.nii.ac.jp/crid/1573950399665672960> (accessed on 7 July 2023).
- LaValle, S.M. *Planning Algorithms*; Cambridge University Press: Cambridge, UK, 2006.
- Kavraki, L.E.; Svestka, P.; Latombe, J.C.; Overmars, M.H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [[CrossRef](#)]
- Effat, H.A.; Hassan, O.A. Designing and evaluation of three alternatives highway routes using the Analytical Hierarchy Process and the least-cost path analysis, application in Sinai Peninsula, Egypt. *Egypt. J. Remote Sens. Space Sci.* **2013**, *16*, 141–151. [[CrossRef](#)]

21. Leidwanger, J. Modeling distance with time in ancient Mediterranean seafaring: A GIS application for the interpretation of maritime connectivity. *J. Archaeol. Sci.* **2013**, *40*, 3302–3308. [[CrossRef](#)]
22. Fu, L.; Sun, D.; Rilett, L.R. Heuristic shortest path algorithms for transportation applications: State of the art. *Comput. Oper. Res.* **2006**, *33*, 3324–3343. [[CrossRef](#)]
23. Antikainen, H. Using the hierarchical pathfinding A\* algorithm in GIS to find paths through rasters with nonuniform traversal cost. *Isprs Int. J.-Geo-Inf.* **2013**, *2*, 996–1014. [[CrossRef](#)]
24. Nepomniashaya, A.S.; Dvoskina, M.A. A simple implementation of Dijkstra’s shortest path algorithm on associative parallel processors. *Fundam. Inform.* **2000**, *43*, 227–243. [[CrossRef](#)]
25. Adoni, W.Y.H.; Nahhal, T.; Aghezzaf, B.; Elbyed, A. The MapReduce-based approach to improve the shortest path computation in large-scale road networks: The case of A\* algorithm. *J. Big Data* **2018**, *5*, 1–24. [[CrossRef](#)]
26. Likhachev, M.; Ferguson, D. Planning long dynamically feasible maneuvers for autonomous vehicles. *Int. J. Robot. Res.* **2009**, *28*, 933–945. [[CrossRef](#)]
27. Botea, A.; Müller, M.; Schaeffer, J. Near optimal hierarchical path-finding. *J. Game Dev.* **2004**, *1*, 1–30.
28. Li, Y.; Su, L.M.; Li, W.L. Hierarchical path-finding based on decision tree. In Proceedings of the Rough Sets and Knowledge Technology: 7th International Conference, RSKT 2012, Chengdu, China, 17–20 August 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 248–256.
29. Jansen, M.; Buro, M. HPA\* enhancements. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Stanford, CA, USA, 6–8 June 2007; Volume 3, pp. 84–87.
30. Harabor, D.; Botea, A. Hierarchical path planning for multi-size agents in heterogeneous environments. In Proceedings of the 2008 IEEE Symposium on Computational Intelligence and Games, Perth, WA, Australia, 15–18 December 2008; pp. 258–265.
31. Sánchez-Ibáñez, J.R.; Pérez-del Pulgar, C.J.; Azkarate, M.; Gerdes, L.; García-Cerezo, A. Dynamic path planning for reconfigurable rovers using a multi-layered grid. *Eng. Appl. Artif. Intell.* **2019**, *86*, 32–42. [[CrossRef](#)]
32. Du, W.; Islam, F.; Likhachev, M. Multi-resolution A\*. In Proceedings of the International Symposium on Combinatorial Search, Vienna, Austria, 26–28 May 2020; Volume 11, pp. 29–37.
33. Tang, Q.; Dou, W. A fast shortest path method based on multi-resolution raster model. In Proceedings of the 2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Xuzhou, China, 16–19 October 2020; pp. 259–262.
34. Etherington, T.R. Least-cost modelling and landscape ecology: Concepts, applications, and opportunities. *Curr. Landsc. Ecol. Rep.* **2016**, *1*, 40–53. [[CrossRef](#)]
35. Fournier, A.; Fussell, D.; Carpenter, L. Computer rendering of stochastic models. *Commun. ACM* **1982**, *25*, 371–384. [[CrossRef](#)]
36. Etherington, T.R.; Holland, E.P.; O’Sullivan, D. NLM py: A python software package for the creation of neutral landscape models within a general numerical framework. *Methods Ecol. Evol.* **2015**, *6*, 164–168. [[CrossRef](#)]
37. Gärds, J.; Oscarsson, M. Exploring the Use of GIS-Based Least-Cost Corridors for Designing Alternative Highway Alignments. 2019. Available online: <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1333968&dswid=3858> (accessed on 7 July 2023).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.