sensors

MDPI

Article

A Grid-Based Approach for Measuring Similarities of Taxi Trajectories

Wei Jiao [1,2], Hongchao Fan [2,*] and Terje Midtbø [2]

[1] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China; jiaowei2017@whu.edu.cn

[2] Department of Civil and Environmental Engineering, Faculty of Engineering, Norwegian University of Science and Technology, 7491 Trondheim, Norway; terjem@ntnu.no

* Correspondence: hongchao.fan@ntnu.no

Received: 13 May 2020; Accepted: 29 May 2020; Published: 31 May 2020

Abstract: Similarity measurement is one of the key tasks in spatial data analysis. It has a great impact on applications i.e., position prediction, mining and analysis of social behavior pattern. Existing methods mainly focus on the exact matching of polylines which result in the trajectories. However, for the applications like travel/drive behavior analysis, even for objects passing by the same route the trajectories are not the same due to the accuracy of positioning and the fact that objects may move on different lanes of the road. Further, in most cases of spatial data mining, locations and sometimes sequences of locations on trajectories are most important, while how objects move from location to location (the exact geometries of trajectories) is of less interest. For the abovementioned situations, the existing approaches cannot work anymore. In this paper, we propose a grid aware approach to convert trajectories into sequences of codes, so that shape details of trajectories are neglected while emphasizing locations where trajectories pass through. Experiments with Shanghai Float Car Data (FCD) show that the proposed method can calculate trajectories with high similarity if these pass through the same locations. In addition, the proposed methods are very efficient since the data volume is considerably reduced when trajectories are converted into grid-codes.

Keywords: similarity measurement; trajectory data; spatial distribution; grid-based approach

1. Introduction

With the development of sensor technology for positioning, ubiquitous Global Navigation Satellite Systems (GNSS) enabled mobile devices to generate huge amounts of trajectory data [1,2]. These trajectory data are sequences containing various information such as location, time, speed and direction, which enables the rapid development of Location-Based Services (LBSs) and applications [3–6]. Similarity measurements, as one of the crucial tasks of data mining, have been widely used in the location-based applications to find all similar trajectories from a large collection [7,8]. Through similarity measurements of trajectory data, valuable information can be mined from large collections of trajectories, such as traffic flows and hot routes [9,10]. It is very useful for many applications, such as urban hotspot detection [11,12], behavior model analysis [13,14], traffic monitoring and prediction [15–17], urban planning [18,19] and location optimization [20,21], etc. Advances in location-based applications are increasingly creating new, sophisticated mechanisms that can foster the exchanges of information among travels to provide better services and promote a sustainable economy.

The information concerned by the similarity calculation is inconsistent in different types of applications. For example, in many sports such as table tennis and football, it is very useful for sport researchers to analyze the movement patterns of top players. The trajectories are classified by using attributes of the trajectory (e.g., direction, speed, curvature, and other descriptors) to find the

similar trajectories of objects' (e.g., players, balls) motions [22,23]. In applications dealing with animal migration patterns and urban emergency, the calculation of similarity needs to focus not only on the spatial location information of the trajectory, but also on the temporal attribute information [24,25]. Further, for location-based spatial trajectories, several applications mainly focus on the location of the routes, while neglecting the time, direction and other information. This is illustrated by the following three different types of application scenarios.

Firstly, in the location recommendation service, the trajectory similarity of moving objects resembles the paths' or locations' similarity. Combing this information, the frequent locations and travel habits can be mined from the large collection of trajectories, and subsequently utilized to serve the ordinary traveler [26–28]. Secondly, in the process of public transportation planning, the similarity measurement can be used by the public transportation company to analyze the aggregated trajectories of residents and obtain typical travel routes (e.g., knowing which routes are accessed densely) [29]. The convenience of public transport can be improved by adjusting the public transport network or adding new routes. The goal is to make the best use of the bus services and maximize the convenience of commuters so as to better serve urban residents. Thirdly, in travel behavior analysis or groups analysis, such as taxi driving behaviors, analyzing the paths' similarity of high-income taxis can provide guidance for ordinary taxi drivers [30,31]. The goal is to reduce the cruising time of taxis and increase revenue. In these kinds of location-based applications, the calculation of trajectories' similarity mainly requires location information, while the speed, direction and time information are not critical. It motivates us to find a suitable method to measure the spatial similarity, which is the important part of the similarity analysis of the spatial dataset.

Methods to measure similarity between trajectories have been well studied on movement pattern related to time sequences. For example, a native similarity measure is the Euclidean distance, calculated as the sum of distances between ordered pairs of sampling points in two trajectories [32]. However, due to the different sampling rates or different speeds in trajectory data, some sampling points may not be well aligned in space. Aiming to overcoming this kind of problem, several algorithms have been proposed [33–35]. For example, the Closest-Pair Distance is used to find the trajectories closest to a given trajectory in a spatial network [36]. The Hausdorff distance is a measure of the similarity between two sets of points [37]. The Dynamic Time Warping (DTW) algorithm allows some sample points to be repeated to achieve optimal alignment [38]. These types of methods struggle with the different sampling rates, trajectory length and the outliers, and achieves good results in trajectory clustering, map matching and other applications. However, such methods focus too much on the details of the trajectory. Hence, small deviations can have a big impact on the results (i.e., causing a large distance between trajectory sequences). Additionally, due to the accuracy of GNSS measurements and the fact that cars may drive on different lanes on the same road, the shape and distance between sequences of the same route may be different, leading to inaccuracy in the calculation results.

Another type of similarity method is calculated based on the road network. The real distance on road network between POIs is used to measure the similarity between the trajectories [1,39,40]. Xia et al. [41] and Abraham et al. [42] mapped the trajectory data to the road network and proposed a new trajectory clustering method, which calculated the similarity of the trajectory by calculating the length of the matching road segment. In applications of spatial data mining based on trajectory data, this type of method is an improvement over previous measurements and can be used for vehicle navigation and route recommendation. However, this kind of method requires detailed road information which are lacking in the vehicle trajectory data. Due to instability in the GNSS signals, errors will inevitably occur during the road matching process (especially in areas with dense road). Subsequently, this may reduce the accuracy of the similarity calculations.

Overall, neither the method of calculating the distance between sequences nor the indirect method of expressing the trajectory in other forms can satisfy the spatial similarity measure of the location-based application in our research. Our key observation is that trajectories are considered similar if they pass through multiple identical locations/places. For example, suppose the triangle area in the Figure 1 is

a square. *T1* describes an object moving on road segment *Road1* and then making a left turn to the road segment *Road2*. *T2* describes an object moving on road segment *Road1*, making a left turn to the road segment *Road3* and then moving on the *Road2* (when the current road is congested, the temporary diversion is selected). Trajectories *T1* and *T2* are similar in location-based studies because they passed through the same locations/places. However, if we consider the distance between sequences, shape or the road information of the trajectory, their similarity is rather small. In this case, it is unnecessary to know the characteristics of the trajectory in detail, although there may be some interesting issues in those trajectories. Hence, the spatial similarity measurement of trajectory data mainly focuses on the common locations or places, and these locations should be the meaningful areas (e.g., a square, commercial streets, office areas), not just points with latitude and longitude. We contend that the location-based similarity analysis is very meaningful, especially if we have some particularly interesting locations for analysis (e.g., travel preferences analysis).

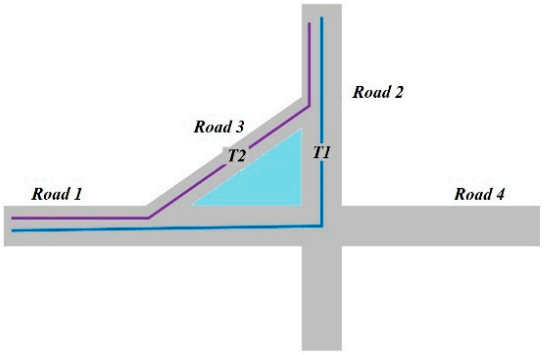

Figure 1. Example of the movement of two vehicles on different road segment.

Due to the facts mentioned above, the existing approaches may be inappropriate in Location-based applications. This paper proposes a grid-based approach called Spatial Grid Coding Distance (SGCD) for measuring similarity of trajectories. Instead of directly using point-list, only grid-codes based on the location of trajectories are selected for similarity measurement. The algorithm converts the trajectories by setting the appropriate grid size, which not only neglects the shape details of the trajectories, but also does not require the alignment between the sampling points. Additionally, the recent development of location-based applications shows that the demand for spatial similarity calculations is becoming more and more diverse. Instead of focusing on one particular model in the process of spatial similarity calculation, we should consider multi-scale query processing [43,44]. The characteristics of common locations and self-intersection of trajectories are considered, respectively. Consequently, two spatial similarity calculation methods are proposed in this paper, including common location similarity and structural similarity. Each kind of similarity measurement algorithm can be used separately during the knowledge discovery process.

The remainder of the paper is organized as follows. Section 2 describes the framework containing the workflow of data processing. The main principles of our approach are introduced in Sections 2.1–2.3. Section 3 develops an experiment in order to test our proposed algorithm. Finally, Section 4 concludes the paper and outline further works.

2. The Grid-Based Approach for Similarity Measurement

A trajectory is a sequence of time-stamped sampling points, and its location information can be represented as a list of geographic coordinates. In order to emphasize the impact of locations while eliminating the disturbance of geometric details of trajectories in the process of similarity measurement, the trajectory was represented with grid-codes in our algorithm. The algorithm runs through two main phases, followed by an algorithmic verification phase (Figure 2). Phase I consists of grid-based trajectory conversion. First, a rule for determining the appropriate grid size is defined in order to

convert the study area into grids (Section 2.1). Second, the original trajectory data (i.e., *Traj-1*, *Traj-2*, . . . , *Traj-n*) is superimposed on a coded grid to achieve trajectory conversion (Section 2.2). After the trajectory conversion, the original trajectory data is represented by grid-codes and converted into *R_Traj-1*, *R_Traj-2*, . . . , *R_Traj-n*. In Phase II, through algorithm design, different types of trajectory similarity measurements are realized in Section 2.3. The following sections explain the workflow in detail.

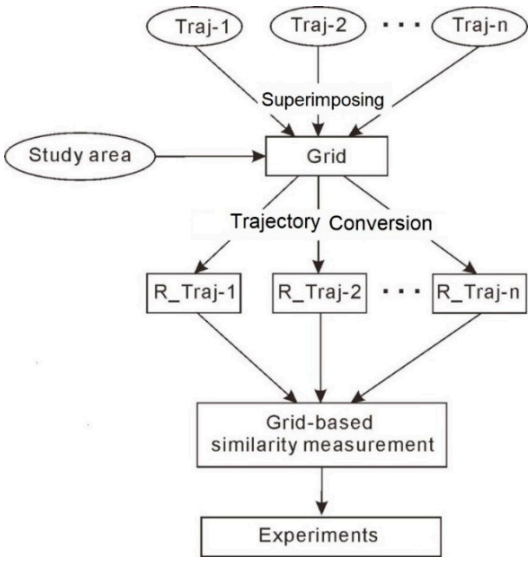

Figure 2. Workflow of the similarity measurement.

2.1. Grid Generation

2.1.1. The Determination of Grid Cell Size

One of the issues of converting the study area from vector feature to regular grid is the determination of the size of the grid cells. If the size is too large, the integrality of trajectory information can be ignored. This will lead to the lack of some basic feature information of trajectories, making the result of the similarity calculation unreliable. If the size is too small, too much attention will be paid to the details of the trajectories. This does not eliminate the effect of the details on the trajectory, leading to small deviations that can have a significant impact on the results. What is more, the computational cost will be very high. Therefore, choosing an appropriate size for the grid cell ensures that the model achieves better performance and acceptable accuracy, considering that the road network in a city is an important indicator of urban development level [45]. Developed areas normally have denser roads and more points of interest (e.g., hospitals, schools), while less developed areas have a sparse road network and fewer meaningful places. Since our research focuses on the spatial similarity calculation based on common locations/places, road network spacing in the city is used as the basis for sizing the grid cells.

However, there are two problems that should be solved in this process. Firstly, the road network in the city can be divided into several levels (e.g., main road, trunk road, secondary trunk road and branch road). Hence, the selection of the level of the road network is the first problem. In general, lower level road networks represent smaller road networks spacing. The lowest level road (i.e., branch road) network will fragment a meaningful place. The blue area in Figure 3 is Square A: the branch road network contains many small paths, which can divide Square A into many fragmented patches (*A1*, *A2*, *A3*, *A4*, *A5*). Further, in similarity analysis, trajectories are normally considered to be similar because they pass by a few identical locations/places. Consequently, the lower-level road network should be selected based on the fact that meaningful places are not fragmented. In this way, the sampling points at the same place can be prevented from being divided into different areas, thereby improving the accuracy of the calculation results.

The second problem is to solve the imbalanced spatial distribution of the road network. Many studies have shown that urban road network density is strongly related to spatial structure [46,47]. Cai et al. used the kernel density estimation to assess the spatial pattern of road density and clearly reflected that the kernel densities of roads decreased with greater distance from urban core areas or central business districts [48]. In fact, many cities have more than one central business district. There are three main theories of urban spatial structure, namely the concentric circle model [49], sector theory [50] and multi-core theory [51]. In terms of the number of urban centers, the urban spatial structure has two types: monocentric and polycentric structure. The density of road networks in downtown areas are higher than in suburban areas in a mono-centered city. In poly-centered cities, densities of the road network in city centers are higher than those in other areas. The road network density is strongly related to the level of urban development, and areas with high road network density have more meaningful locations. Since our research is based on spatial similarity calculation for meaningful locations, the number of locations in each grid cell should be balanced. For this reason, in the process of determining the grid size, the areas with high road network density and the areas with low road network density should be treated separately.

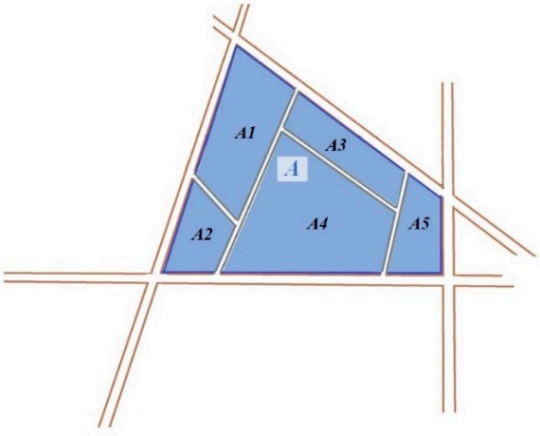

Figure 3. Example of branch road in a garden.

According to the road network density, the study area can be divided into several sub-areas (such as $R_1, R_2, \ldots, R_i$). Generally, a city can be divided into suburban and downtown areas. The kernel density estimation is used to calculate the road network density in the sub-region R_i. In the kernel density analysis, the bandwidth is an important factor which directly affects the results of the density analysis. This study used the calculation formula proposed by Silverman [52] (Equation (1)). A raster data of road network density is generated by kernel density calculation, and its attribute value is the road network density within the range of cells. The cell can be called a density unit in our research.

$$bandwidth = 0.9 * \min\left(SD, \sqrt{\frac{1}{\ln(2)}} * D_m\right) * n^{-0.2}. \quad (1)$$

where SD is the standard deviation, D_m is the median distance. This formula weighs the two parameters of standard deviation and median distance and takes the minimum of the two to contribute to the final calculation.

According to the relationship between the road network density and the road network spacing, the average value of the road network spacing in the sub-region R_i is calculated as the grid size of this area. So, the grid size G ($road_{level}$, R_i) can be calculated according to the following equation:

$$G(road_{level}, R_i) = \frac{\sum_{j=0}^{m_i} f(road_{level}, R_i, j)}{m_i}, \quad (2)$$

where the f ($road_{level}$, R_i, j) is the road network spacing in the *j-th* density unit of raster data (kernel density analysis result) in sub-area R_i; the $road_{level}$ is the selected level of road network; m_i is the number of density unit in subarea R_i. According to Miyagawa [53], f ($road_{level}$, R_i, j) can be calculated according to the following equation:

$$f(road_{level}, R_i, j) = \frac{2}{density\_road(road_{level}, R_i, j)}. \quad (3)$$

where the *density_road* ($road_{level}$, R_i, j) is the density of the selected road network in the *j-th* density unit of sub-area R_i.

2.1.2. The Generation of Grid

For an irregular convex polygonal study area, in order to divide it into a regular grid, a Minimum Bounding Rectangle (MBR) needs to be established. The MBR is a rectangle oriented to the x and y axes and it is one of the most frequently used methods to express the geographic feature or a geographic dataset. The MBR is determined by two coordinates: *Xmin*, *Ymin* and *Xmax*, *Ymax*, which are obtained based on the geographic coordinate range of the study area. Assuming there exists a study area, the length of the MBR is x and the width is y. According to the method proposed in the previous section, the size of the grid cell of the study area is determined as δ. The conversion of the grid is as follows:

$$\mu_x = \mu_y = \delta, \quad (4)$$

$$M = \frac{x}{\mu_x}, \quad (5)$$

$$N = \frac{y}{\mu_y}. \quad (6)$$

where the μ_x, μ_y are the length and width of each grid cell respectively; M and N are the numbers of the rows and columns, respectively.

According to the previous section, the study area should be divided into several sub-areas to solve the problem of spatial heterogeneity. Subsequently, separate grids are generated for each sub area before they are all merged into one grid.

2.2. Converting Trajectory with Grid Code

Let T be a trajectory in space, represented as T = ($p1$, $p2$, . . . , pn), where n is the number of sample points in T and pi ($1 < i < n$) is a sample point with geographic coordinates. These sample points are arranged in the order of sample time.

Since this study focuses on the spatial similarity of trajectory data, we only need to pay attention to the location information of the trajectory during the trajectory conversion process. As shown in the Figure 4, the trajectory data is superimposed on the coded grid of the study area. If the sample point is within the grid cell, the location information of the sample point is replaced with the grid code. By converting each sample point, the trajectory is finally converted into grid-codes.

It is worth mentioning that, for the vehicle trajectory data, the sampling frequency is 10–60 s, which may result in multiple consecutive sampling points falling in the same grid cell. To reduce

data redundancy, duplicate data needs to be filtered (i.e., $num_{i+1} \neq num_i$). Through the filtering of the trajectory it is possible to both reduce the redundancy and maintain the integrity of the data. This is beneficial for the data storage. The converted trajectory is shown in Figure 4. The attribute value of grid cell is the number of times that the trajectory passes through the grid cell. For grid (3,4), the attribute of the grid is five, which means that the trajectory passes through this grid cell five times. After the trajectory conversion, the location information of trajectory is expressed by the grid-codes and the spatial distribution of the trajectory is represented by the trajectory matrix.

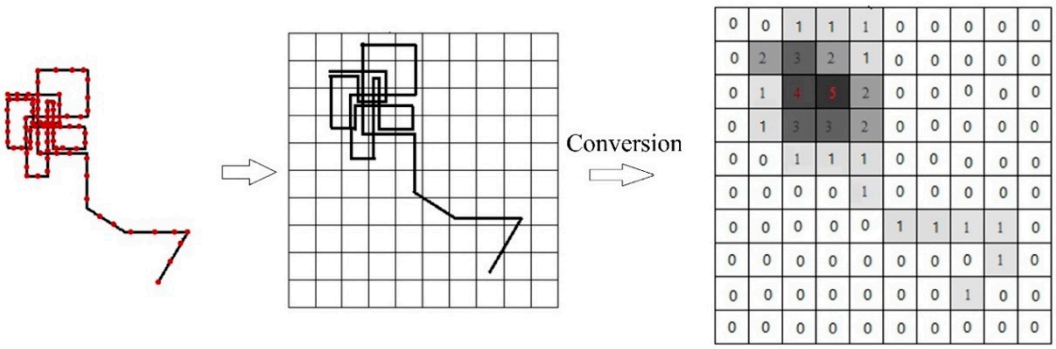

Figure 4. An example of trajectory conversion.

2.3. The Grid-Based Similarity Measurement

Through the conversion of trajectories, a matrix with grid-codes is generated to represent a trajectory. The trajectory similarity calculation can be done by comparing and analyzing codes in the two matrices. In this study, there are mainly two types of spatial similarity measures. These are the common-locations similarity, which consider the common locations visited by the two trajectories, and the structural similarity, which considers the self-intersection of a trajectory.

For the first category, the similarity of the common location of the trajectory mainly focuses on the common locations a moving object has visited. This may be useful for location recommendation services, or for finding objects which move through certain points of common interest (e.g., emergency locations, terrorist locations, etc.).

However, in many location-based applications it is not enough to consider only the common locations of two trajectories. Figure 5 shows an example with two taxi trajectories in Shanghai. The taxi activities are considerably restricted by geographical space, and the range of the two trajectories varies a lot. Since both trajectories in Figure 5 include the route to the airport, if only the common locations are considered the similarity between the two trajectories will be high. This result is obviously unreliable in the analysis of behavior pattern. Consequently, it is very important to consider repetitive and self-intersecting features of trajectories in the applications of movement patterns (e.g., analysis of travel behavior preference, hotspot extraction). As for the second category, the structural similarity algorithm has been designed. It not only focuses on the common points of interest that trajectories pass by, but also on the information of repetitive active regions and self-intersections of trajectories.

The following sections explain the measurements in detail.

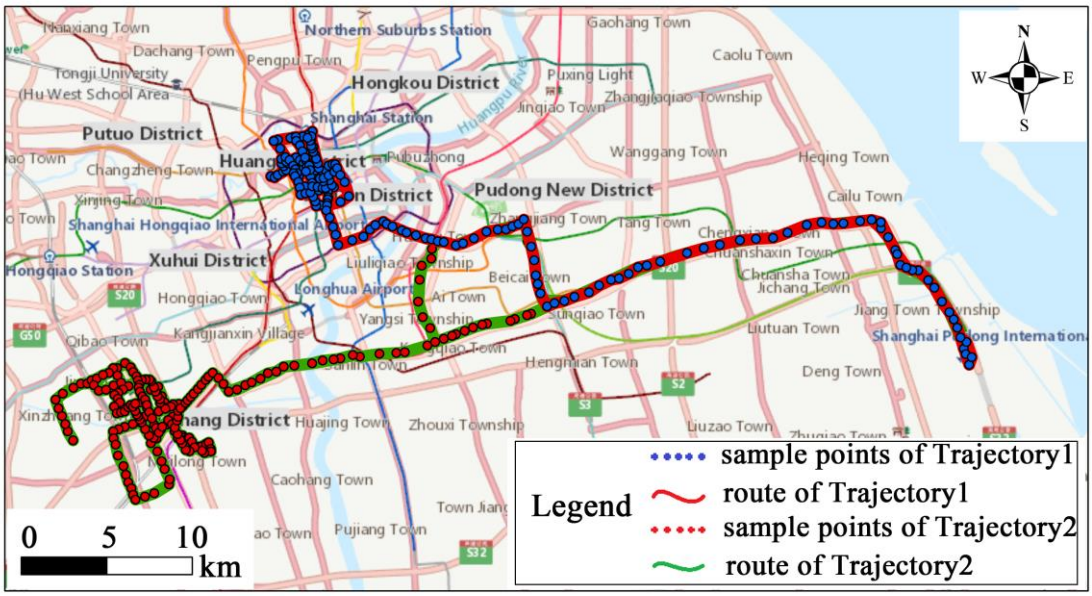

Figure 5. The trajectories of two taxis.

2.3.1. Similarity of Common Locations

A spatial similarity measurement (Sim_{loc} (T_i,T_j)), focusing on common locations in trajectories, is introduced in this subsection. It counts the number of the common locations and calculates its percentage of all location points in trajectories as the similarity of the two trajectories. Note that in the process of calculating the similarity of common location points, we focus on whether a location point is passed or not, regardless of how many times it passes. Therefore, before calculating the similarity of common locations, we need to do unique value processing for each track.

Let T_i and T_j be two trajectories. After trajectory conversion, T_i and T_j are converted into two trajectory matrices (T_matrix_i, T_matrix_j). The spatial similarity measure between these two trajectories is defined based on the common elements in these trajectory matrices.

$$Matrix_i = f(T\_matrix_i), \tag{7}$$

$$Matrix_{common} = Matrix_i \cap Matrix_j, \tag{8}$$

$$Matrix_{total} = Matrix_i \cup Matrix_j, \tag{9}$$

$$Sim_{loc}\left(T_i, T_j\right) = \frac{sum(Matrix_{common})}{sum(Matrix_{total})}. \tag{10}$$

where f (*) represents a function of the unique value calculation; $Matrix_i$ is the grid-coded trajectory matrix after the unique value calculation; $Matrix_{common}$ is the matrix of the common locations; $Matrix_{total}$ is the matrix of the total locations of the trajectory i, and j. Sim_{loc} (T_i,T_j) is the ratio of the sum of the two matrix values.

Please note that the above similarity measure satisfies the following properties:

1. Sim_{loc} (T_i,T_j) >= 0.
2. Sim_{loc} (T_i,T_j) = Sim_{loc} (T_j,T_i).
3. Sim_{loc} (T_i,T_j) belongs to [0,1].

In Figure 6, the similarity calculation of the taxi trajectories *T1* and *T2* is visualized. After the trajectory conversion and the unique value processing, trajectory matrices are obtained and displayed in the grid, where 1 indicates that the trajectory passed the grid and 0 indicates that it did not pass. After calculation, *T1* passes through 27 grid cells, *T2* passes through 24 grid cells. The intersection of

trajectory matrices (see the common part) is 18, and the union is 27 + 24 − 18 = 33. So, the result of the similarity is 18/33 = 0.55.

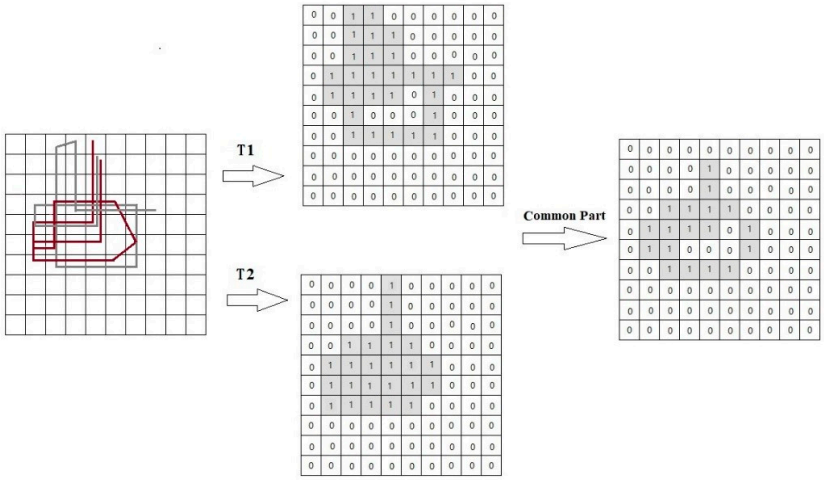

Figure 6. Example of common locations similarity measurement.

2.3.2. Structural Similarity Measurement in Trajectories

The structural similarity measurement (Sim_{str} (T_i,T_j)) focuses on the information of repetitive active regions and self-intersections in trajectories. It mainly focuses on the number of visits in the common locations of the trajectories and can help to extract the behavior preferences of moving objects in the analysis of movement patterns. The structural similarity of the two trajectories is defined as:

$$Matrix_{common} = \text{minimum}(Matrix_i, Matrix_j), \tag{11}$$

$$Matrix_{total} = Matrix_i + Matrix_j - Matrix_{common}, \tag{12}$$

$$Sim_{str}\left(T_i, T_j\right) = \frac{sum(Matrix_{common})}{sum(Matrix_{total})}, \tag{13}$$

where $Matrix_i$ is a trajectory matrix whose values are the number of times the grid cell has been visited. $Matrix_{common}$ is a matrix whose values represent the common times of the trajectory i and j to visit this grid cell. $Matrix_{total}$ is the union matrix of two matrix, and its values are obtained by subtracting the minimum visited times from the total visited times of the trajectory i, and i. Sim_{str} (T_i,T_j) is the ratio of the sum of the two matrix values.

Please note that the above similarity measure satisfies the following properties:

1. Sim_{str} (T_i,T_j) >= 0.
2. Sim_{str} (T_i,T_j) = Sim_{str} (T_j,T_i).
3. Sim_{str} (T_i,T_j) belongs to [0,1].

Using this algorithm, the process of calculating the structural similarity of trajectories *T1* and *T2* is shown in Figure 7. The attribute value represents the number of times the trajectory passes through the grid cell. For *T1* and *T2*, the sum of the grid attributes of the common part is 21, and the sum of these two trajectory matrices are 34 and 32, respectively. So, the result of the similarity is 21/(34 + 32 − 21) = 0.47.

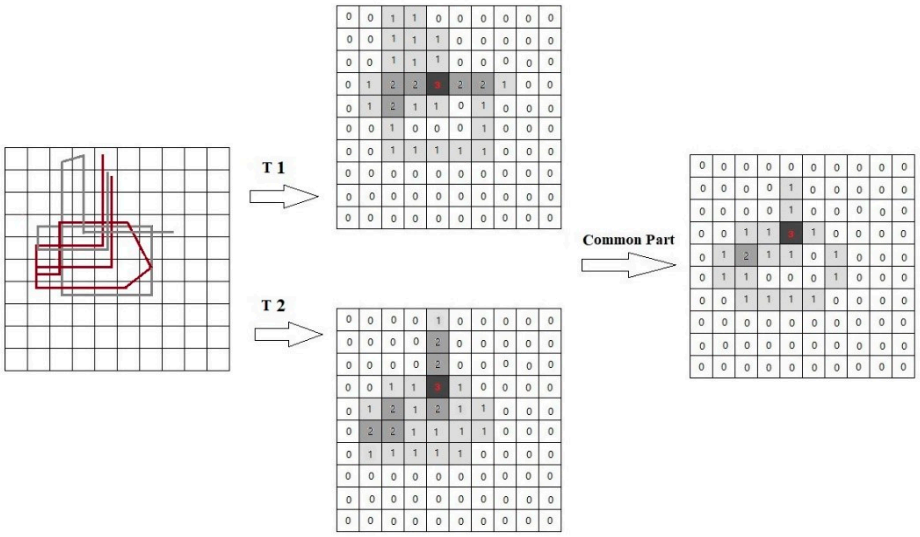

Figure 7. Example of structural similarity measurement.

2.3.3. The Combined Spatial Similarity

There are some applications that need to consider these two similarities comprehensively. For example, taxi trajectories data, as a spatial trajectory data for profit, are mainly distributed in the down areas with a lot of pick-up/drop-off points. By calculating the structural similarity of the trajectories, a similarity set with a large number of trajectories can be extracted, which is mainly distributed in the city center. In order to explore the travel patterns within the similar set, it is necessary to refine the trajectory using the similarity of the common locations. For spatial similarity analysis of this type of application, a comprehensive metric is needed. It can be used as the distance measure to subdivide similar sets and further mine their spatial characteristics.

Xia et al. [41] considered that spatial and temporal characteristics have the same weight in the spatiotemporal similarity calculation of trajectories. Abraham et al. [42] transformed the trajectory into binary code and believe that location, sequence and other factors have the same effect on the combined similarity. Consequently, in the process of calculating the combined spatial similarity, our research considers that the common location similarity and the structural similarity have the same weight. The combined spatial similarity measure is obtained by:

$$Sim\left(T_i, T_j\right) = \left(Sim_{loc} + Sim_{str}\right)/2, \tag{14}$$

The result also satisfies the following properties:

1. $Sim\ (T_i,T_j) >= 0$.
2. $Sim\ (T_i,T_j) = Sim\ (T_j,T_i)$.
3. $Sim\ (T_i,T_j)$ belongs to [0,1].

In the end, the similarity of *T1* and *T2* is (0.55 + 0.47)/2 = 0.51.

Note that each kind of similarity measurement algorithm can be used separately according to different application scenarios and needs. The applicable scenarios have been illustrated in the corresponding subsections.

3. Experiments

We have implemented our method and conducted an extensive set of experimental studies in order to (1) test the proposed techniques; and (2) compare our method with another model. In this section, a case study is conducted using taxi trajectory data from Shanghai, China (all administrative districts except Chongming district). Chongming District is a suburban county in Shanghai far away

from the city area. The taxis in this area are restricted by the natural environment, resulting in rare contact with the mainland of Shanghai. According to the statistics of our floating car data, less than 1% of the data is related to Chongming District. Since this section is to verify and evaluate our algorithm, Shanghai city area with more trajectories was selected as the research area. The method used is the combined spatial similarity algorithm proposed in our research. The experiments were conducted on an Intel Core i5-7500 Quad-core machine running Windows 10 with 16 GB of RAM and a 250 GB SATA2 512-MB hard drive.

3.1. Data Preprocessing and Experimental Setup

3.1.1. The Trajectory Data Description

The taxi trajectory data used in this paper were provided by a commercial company in Shanghai. They are temporally ordered position records collected from about 6000 Global Positioning System (GPS) enabled taxis. The data was collected over a period of 7 days from 4 June to 10 June 2018. The average sampling interval of the data was 10 s. In the database, the trajectory data of each vehicle is a series of position records arranged in chronological order. Each record has many attributes, i.e., Taxi identifier Time, Speed, Direction, Current location (longitude, latitude) and Passenger state. The "Taxi identifier" is a unique identifier of a taxi, while "Time" contains an accurate date and time for each record. "Speed" represents the speed of a taxi at a given time. "Direction" is the horizontal angle measured clockwise from the north direction. "Passenger state" is a Boolean variable that denotes whether the taxi is carrying passengers or not. In space, the raw GPS trajectory data is represented by a series of discrete points. Figure 8 illustrates the sample points of one taxi with the identification number of 10,383 on the 4 June 2018. The sample points are red, while the trajectory route is green. The trajectory route is generated by sample points in chronological order.

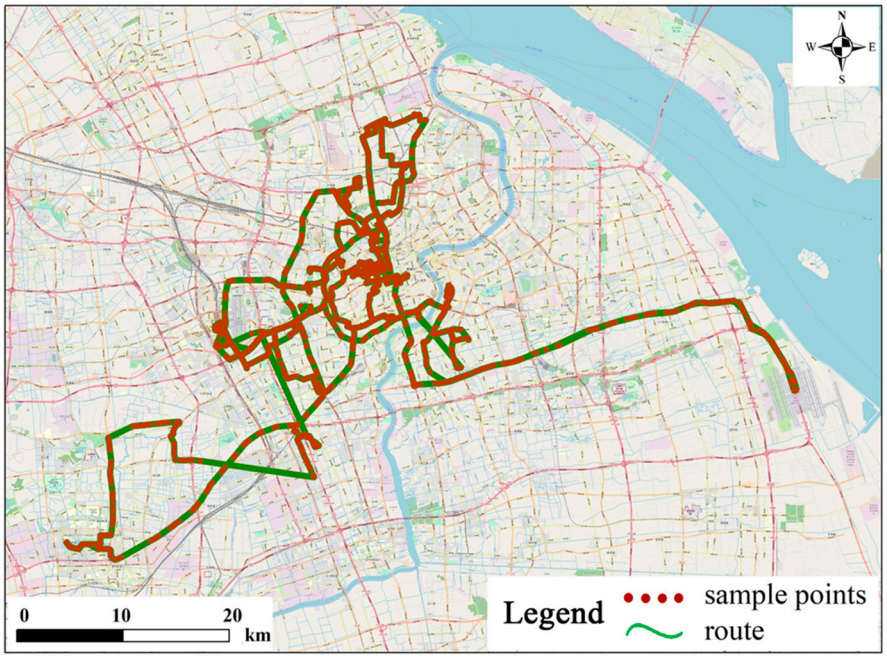

Figure 8. The trajectory of a taxi with a taxi identifier of 10,383 on the 4 June 2018.

3.1.2. Grid Generation

In our example, the road network of Shanghai was used to determine the size of the grid cell, and subsequently utilized to generate the grid for Shanghai. The roads were obtained from OpenStreetMap (OSM) (http://www.openstreetmap.org) because it is freely available [54] and has comparable quality to the authority data based on our local knowledge and the existing study [55].

The roads in Shanghai can be divided into five levels. These are expressways, main roads, trunk roads, secondary trunk roads and branch roads. In this study, the method of visual interpretation was adopted to select the suitable road network. For example, in the blue frame (Figure 9), the branch road network contains too many small paths, which can divide the "Town God's Temple" area into many fragmented patches. Consequently, the second trunk and above road (i.e., expect branch roads) should be selected as input data.

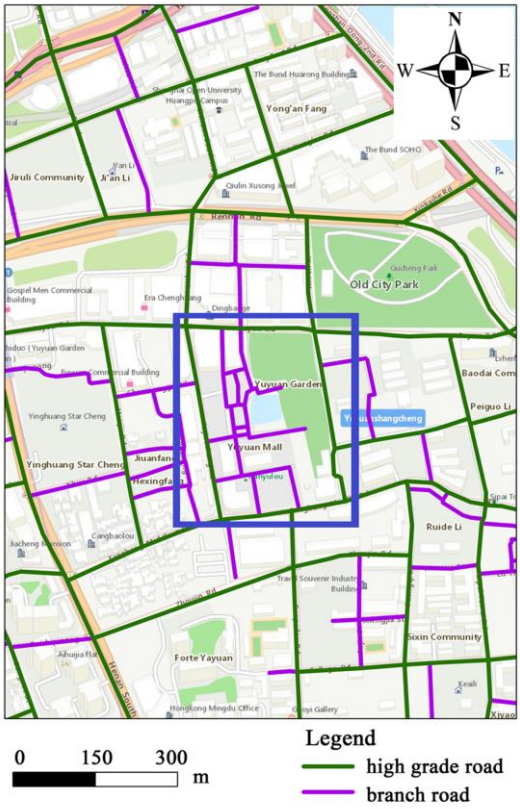

Figure 9. Visualization of different grade road network.

The spatial distribution of the road network in Shanghai is unbalanced (as shown in Figure 10a,b). The density distribution of the road network coincides with the ring road (Figure 10b), which are the two ring expressways. The road network density in the inner ring area is the highest, successively, in the area between the inner and outer rings and in the outer ring area. Therefore, the study area was divided into three sub regions. These are the inner ring area, the area between the inner and outer rings, and the outer ring area. The road network spacing per unit area of the three regions was calculated, respectively. As shown in the histogram (Figure 10c), the average distance of the road spacing in different zones varied a lot. The average distance in the inner ring area is 350 m (mainly ranging from 200 to 450 m), followed by the 700 m in the area between the inner and outer rings (mainly ranging from 400 to 1000 m) and 1500 m in the outer ring area (mainly ranging from 800 to 2000 m). Consequently, the grid size of the three sub areas is 350, 700 and 1500, respectively (Figure 10d).

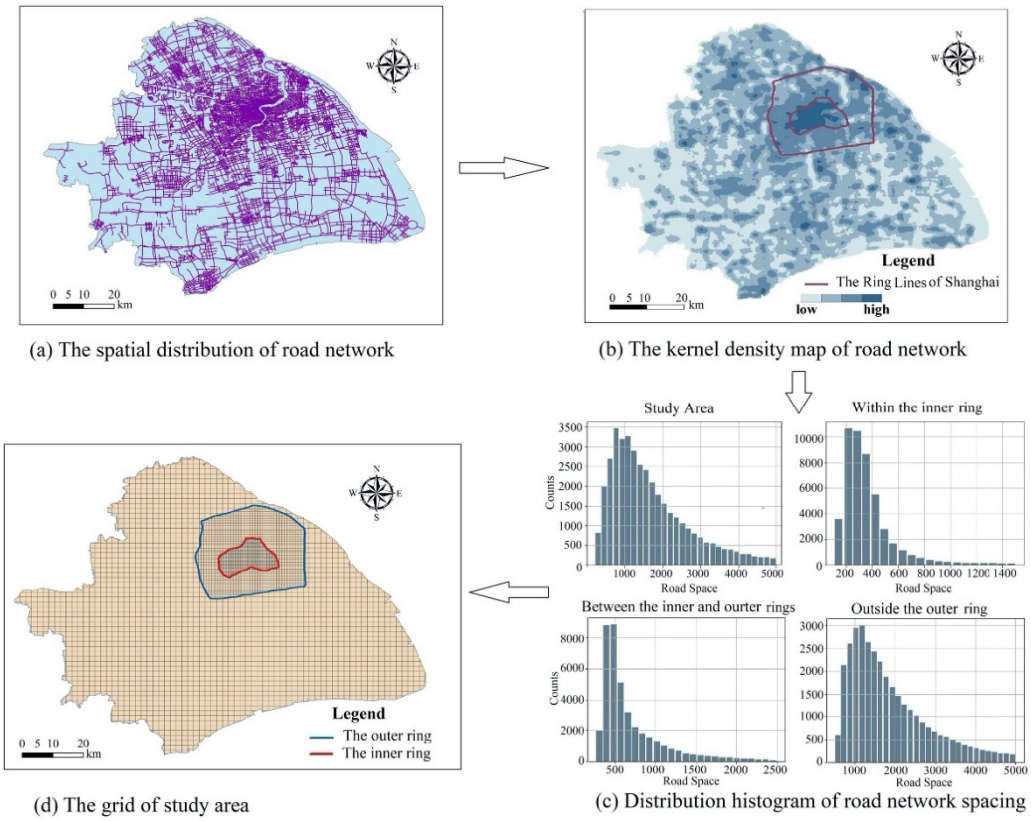

Figure 10. Grid size determination procedure.

3.2. *Effectiveness of the Method*

In order to evaluate the effectiveness of our approaches, we randomly selected 2000 taxi trajectories (i.e., 1000 pairs) from our dataset. Each trajectory contained at least 200 sampling points. By visualizing the trajectories on the digital map, similar trajectory pairs were manually labeled as the ground truth. A total of 72 trajectories were labeled in this experiment. The precision and recall of the similar pairs in different similarity threshold were used in this section. The precision is the portion of real similar pairs (as indicated by the ground truth) in all similar trajectory pairs found by the method in our study (i.e., Precision = Real similar trajectories/Similar trajectories). The recall is the ratio of the number of the real similar trajectory pairs in the results calculated in this case to the number of real similar trajectory pairs in the ground truth (i.e., Recall = Real similar trajectories/ground truth).

The Table 1 shows the results of detecting similar pairs of trajectories with different similarity thresholds, where the term of Real Similar Trajectories denotes the similar pairs correctly identified in our method. With the increase of the Similarity Threshold (from 0.6 to 0.8), the precision was significantly improved, while the recall decreased slightly. This is because the higher the similarity threshold, the fewer similar trajectories can be found, which reduces the denominator of precision and the numerator of recall. As shown in Table 1, when the similarity threshold is greater than 0.8, the precision and recall are 0.969 and 0.861, respectively. This performance of our approach can meet the needs of data mining, especially for trajectories with different geographic coordinates and shapes. As shown in Figure 11, the similarity of the two trajectories (i.e., T1 and T2) is 0.85. The exact geographic coordinates and the shape of the two trajectories (T1 and T2) are different, but the main places that the two trajectories passed through are consistent (e.g., the area of Shanghai General Hospital, Yu Garden). Therefore, the two trajectories are considered to be similar.

Table 1. The results of similar pairs detecting.

Similarity Threshold	Similar Trajectories	Real Similar Trajectories	Precision	Recall
0.6	214	72	0.336	1
0.7	130	70	0.538	0.972
0.8	64	62	0.969	0.861

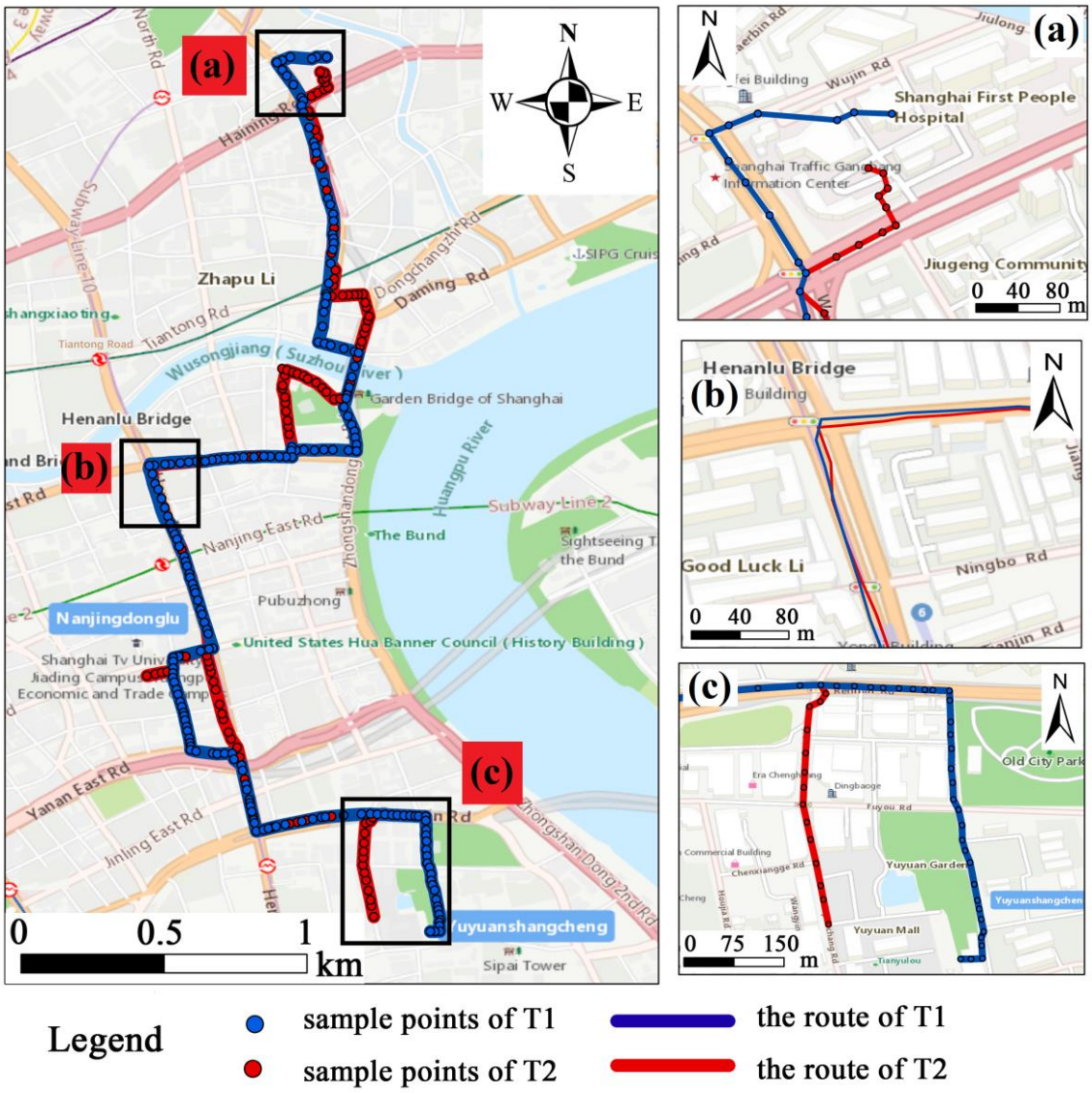

Figure 11. Example of two similar trajectories.

As shown in Figure 11a, these sampling points of the two trajectories belong to the same area, Shanghai General Hospital. However, due to the different locations of the pick-up points, the geographic coordinates and shapes of the two trajectories in this area are different. The area in Figure 11c is a very famous scenic spot named Yu Garden, where the density of vehicles and population is high. Vehicles are likely to choose other routes to reach this area due to the road congestion. In our approach, we focus on the place where the trajectory passes rather than the geographic coordinates. Therefore, the trajectories to reach this area are considered similar. Additionally, due to the unstable GPS signal, the geographic coordinates of the sampling points will have an error of 10–20 m. The shape and distance between sequences of the same route may be different (Figure 11b), which may cause the calculation results to be inaccurate.

In all, unstable GPS signal, vehicle lane change, traffic congestion and the past travel experience can lead to different trajectory shapes and road information. However, the main locations/places they pass through are consistent, so the two trajectories should be considered very similar. In our method, shape details and the road information of trajectories are neglected while emphasizing the integrity of the trajectory. Consequently, the algorithm proposed in this paper has good robustness.

According to this feature, the trajectory similarity measurement proposed in this paper can be well applied in location recommendation systems or the group analysis mentioned in the introduction. This is because the calculation of trajectories' similarity in these methods mainly pay attention to whether the trajectory reaches a specific area. In addition, based on the similarity results calculated in this paper, typical routes can be extracted from the mass trajectories to provide guidance for travel route recommendations and route planning of urban public transportation systems. For verification, we extracted the trajectories of Pudong International Airport and calculated their similarities. Then, as a traditional density-based clustering algorithm, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm was used to visualize our similarity calculation results. The DBSCAN algorithm requires two parameters: ε (*eps*), and the minimum number of points required to form a high-density area (*minPts*) [56]. It starts with an arbitrary unvisited point and then explores the ε-neighborhood of this point. If there are enough points in the ε-neighborhood, a new cluster is established, otherwise the point is labeled as noise.

Figure 12a illustrates the 947 visit trips on 4 June 2018 (excluding the departure trajectories from the airport). After calculating the similarity of trajectories, the DBSCAN algorithm was used to cluster the airport trips, but the distance parameter in the DBSCAN algorithm was replaced by trajectory similarity. In the DBSCAN algorithm, the parameters were set as $esp = 0.8$, $minpts = 10$, i.e., the similarity of trajectories is more than 0.8, and the number of trajectories in the ε-neighborhood is more than 10. The DBSCAN algorithm can remove some noise trajectories with low similarity and divide airport trajectories into nine categories based on parameter settings. The visualization results are shown in the Figure 12b. By utilizing the similarity results calculated in our study, meaningful trajectory clusters and typical routes can be generated.

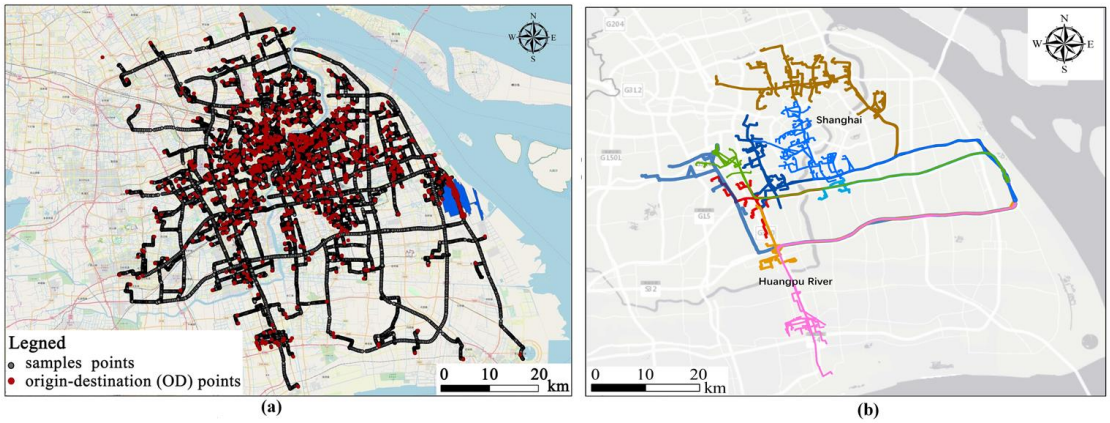

Figure 12. (**a**) Airport visit trips of Pudong international airport on 4 June 2018. (**b**) Clustering results of Pudong airport travel based on similarity. The nine colored lines indicate the categories divided by DBSCSN based on trajectories similarity.

3.3. Comparison with State-of-the-Art Model

The performance of our model is evaluated by comparing it with Fast Dynamic Time Warping (FastDTW), which is an approximate Dynamic Time Warping (DTW) algorithm that provides optimal or near-optimal alignments with an O(N) time and memory complexity [57]. The DTW algorithm is widely used in similarity calculations and its performance has been verified by several authors [38,58,59]. However, the quadratic time and space complexity of DTW is limited to small time series datasets.

FastDTW can, on the one hand, be run on much larger data sets. It is also an order of magnitude faster than DTW. Consequently, the FastDTW algorithm is used for model efficiency comparison experiments, including data storage and the elapsed time of the similarity calculation. The experiments were conducted by using the same data set with opensource codes FastDTW https://pypi.org/project/fastdtw/ and DTW https://pypi.org/project/dtw/.

We selected 10 sets of data for the experiment, and the number of trajectories was 50, 100, . . . , 500, respectively. For the data storage, not only the sizes of data space but also the number of trajectory records were compared between the original trajectory and the converted trajectory. As shown in the Figure 13, the data space for original and converted data changed linearly with the increased number of trajectories. The comparison shows a significant difference between the original trajectory and the converted trajectory. The original trajectory occupies 34-times more storage space than the converted. By transforming and converting the trajectory data, the record of the data is only 1/10 of the original data.

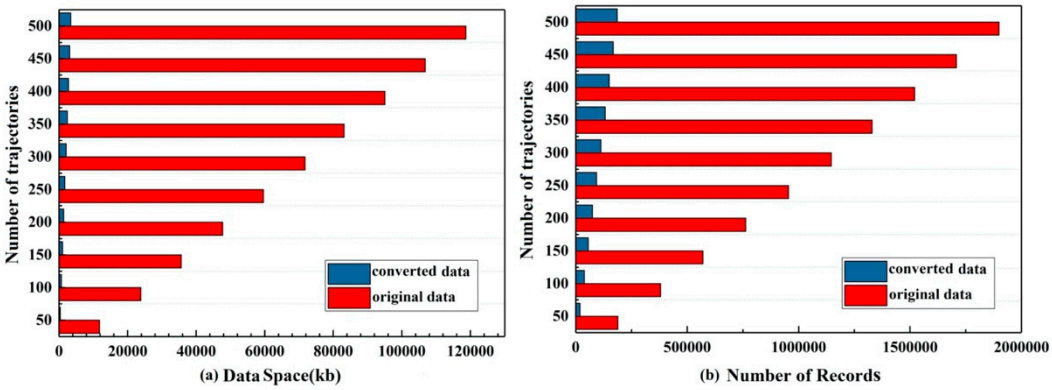

Figure 13. Comparison of the (**a**) data space (kb) and (**b**) the number records for the trajectory dataset.

Subsequently, the performance of the similarity search technique between our algorithm and FastDTW algorithm is measured by comparing the average elapsed time. As shown in Figure 14, the experimental results confirm that the elapsed time of the two models increases linearly as the number of trajectories increase. However, the time consumption for the FastDTW algorithm is higher than our method (about 2.4 times). There are two main reasons. First, although our model needs a part of time for trajectory conversion, trajectory conversion can greatly reduce the amount of data to reduce the elapsed time. Second, our model does not need to calculate the distance between each pair of sampling points. Therefore, our proposed similarity algorithm (SGCD) has great advantage in processing large-scale trajectory data.

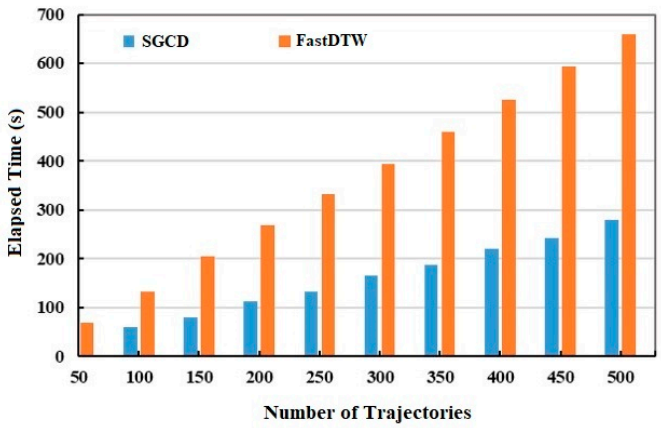

Figure 14. Comparison of the elapsed times for the trajectories.

In addition, we use the FastDTW model to calculate the similarity of 1000 pairs of trajectories used in Section 3.2. The result of the FastDTW model is the sum of the shortest distance between aligned sampling points. The smaller the distance between the trajectories, the more similar they are. For each distance function d, we can define its associated similarity function as $s(x, y) = 1/(1 + d(x, y))$, which ranges from 0–1. Note that property should be dropped to avoid dimensional problems [60,61].

The calculation results are shown in Table 2. With the increase of the similarity threshold, the precision of the FastDTW model is acceptable, but the data recall rate and the number of correct trajectories identified are very low. There are two main reasons. First, it is related to the transformation of distance function and similarity function in the calculation process, and the outliers have a greater impact on the calculation results. Secondly, the structural similarity of trajectories is considered in our model, and the principle of the two models and the definition of similar trajectories are different. Therefore, the FastDTW model is not suitable for the location-based trajectory similarity research mentioned in this study.

Table 2. The results of similar pairs detecting by Fast Dynamic Time Warping (FastDTW).

Similarity Threshold	Similar Trajectories	Real Similar Trajectories	Precision	Recall
0.6	26	20	0.769	0.278
0.7	20	16	0.80	0.222
0.8	14	10	0.714	0.139

4. Conclusions and Future Work

In this paper, an algorithm of Spatial Grid Coding Distance (SGCD) was designed to calculate the trajectories' similarity. Instead of relying on aligned sample points as in traditional approaches, it can use the grid to convert the trajectory data and identify the similar trajectories passing through the same places. In order to obtain better performance and acceptable accuracy, a rule for determining the appropriate grid size is developed by calculating the network spacing. By considering the self-intersection characteristics of trajectories, two similarity calculation algorithms are designed: namely, the common-locations similarity and structural similarity. Experimental study on real datasets verified the advantages and efficiency of our algorithm.

The similarity calculation of vehicle trajectory can be used in many traffic-related applications. For example, in traffic congestion recognition applications, our algorithm can be adjusted by considering the number of consecutive sampling points in each grid and subsequently utilizing them to identify the congestion area. In addition, in the field of human sociology, its application prospect is also widely concerned (e.g., behavior pattern analysis, service sharing). Note that taxis are one of the important ways to travel. If data for all types of travel could be obtained (e.g., private cars), more valuable information will be mined. For example, the movement of private cars is more regular than that of taxis. In general, the main activity on weekdays is commuting (especially at specific times of the day). By calculating the common location similarity and structural similarity of the trajectory, information such as the main active area and routes of the trajectory can be extracted more quickly and accurately. This would be helpful for the development of behavior pattern analysis and carsharing services. As a continuation, we plan to use the algorithm of similarity measurement to extract semantic information from different types of trajectories and further explore human behavior patterns. Some other data, such as weather factors, regional terrain factors and social media data, may be added to improve the accuracy of the results.

Author Contributions: H.F. had the idea and designed the concept of implementation and test. W.J. did the data analysis, conducted all the implementation, and wrote the draft of this paper. T.M. contributed comments to improve the method and revised the manuscript in terms of language. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the NSFC (National Natural Science Foundation of China) project No. 41771484. The authors gratefully acknowledge financial support from China Scholarship Council.

Conflicts of Interest: All authors declare that they have no conflicts of interest.

References

1. Han, B.; Liu, L.; Omiecinski, E. Road-network aware trajectory clustering: Integrating locality, flow, and density. *IEEE Trans. Mob. Comput.* **2013**, *14*, 416–429.
2. Kong, X.; Liu, Y.; Wang, Y.; Tong, D.; Zhang, J. Investigating public facility characteristics from a spatial interaction perspective: A case study of Beijing hospitals using taxi data. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 38. [CrossRef]
3. Liu, K.; Gao, S.; Lu, F. Identifying spatial interaction patterns of vehicle movements on urban road networks by topic modelling. *Comput. Environ. Urban Syst.* **2019**, *74*, 50–61. [CrossRef]
4. Delafontaine, M.; Neutens, T.; Van de Weghe, N. A GIS toolkit for measuring and mapping space–time accessibility from a place-based perspective. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 1131–1154. [CrossRef]
5. Van Weerdenburg, D.; Scheider, S.; Adams, B.; Spierings, B.; van der Zee, E. Where to go and what to do: Extracting leisure activity potentials from Web data on urban space. *Comput. Environ. Urban Syst.* **2019**, *73*, 143–156. [CrossRef]
6. Zhou, X.; Chen, X.; Kimmons, B. Detecting tourism destinations using scalable geospatial analysis based on cloud computing platform. *Comput. Environ. Urban Syst.* **2015**, *54*, 144–153. [CrossRef]
7. Li, M.; Ye, X.; Zhang, S.; Tang, X.; Shen, Z. A framework of comparative urban trajectory analysis. *Environ. Plan. B Urban Anal. City Sci.* **2018**, *45*, 489–507. [CrossRef]
8. Gao, S.; Wang, Y.; Gao, Y.; Liu, Y. Understanding urban traffic-flow characteristics: A rethinking of betweenness centrality. *Environ. Plan. B Plan. Des.* **2013**, *40*, 135–153. [CrossRef]
9. Cao, J.; Wu, Z.; Wu, J. Scaling up cosine interesting pattern discovery: A depth-first method. *Inf. Sci.* **2014**, *266*, 31–46. [CrossRef]
10. Lieske, S.N.; Leao, S.Z.; Conrow, L.; Pettit, C. Assessing geographical representativeness of crowdsourced urban mobility data: An empirical investigation of Australian bicycling. *Environ. Plan. B Urban Anal. City Sci.* **2019**. [CrossRef]
11. Alexander, L.; Jiang, S.; Murga, M.; González, M.C. Origin–destination trips by purpose and time of day inferred from mobile phone data. *Transp. Res. Part C Emerg. Technol.* **2015**, *58*, 240–250. [CrossRef]
12. Wu, H.; Fan, H.; Wu, S. Exploring Spatiotemporal Patterns of Long-Distance Taxi Rides in Shanghai. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 339. [CrossRef]
13. Kim, J.; Mahmassani, H.S. Spatial and temporal characterization of travel patterns in a traffic network using vehicle trajectories. *Transp. Res. Procedia* **2015**, *9*, 164–184. [CrossRef]
14. Siła-Nowicka, K.; Vandrol, J.; Oshan, T.; Long, J.A.; Demšar, U.; Fotheringham, A.S. Analysis of human mobility patterns from GPS trajectories and contextual information. *Int. J. Geog. Inf. Sci.* **2016**, *30*, 881–906. [CrossRef]
15. Wang, Q.; Lu, M.; Li, Q. Interactive, multiscale urban-traffic pattern exploration leveraging massive GPS trajectories. *Sensors* **2020**, *20*, 1084. [CrossRef] [PubMed]
16. Park, Y.; Mount, J.; Liu, L.; Xiao, N.; Miller, H.J. Assessing public transit performance using real-time data: Spatiotemporal patterns of bus operation delays in Columbus, Ohio, USA. *Int. J. Geogr. Inf. Sci.* **2019**, *34*, 1–26. [CrossRef]
17. Mohan, P.; Padmanabhan, V.N.; Ramjee, R. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, Raleigh, NC, USA, 16–19 November 2008; pp. 323–336.
18. Levy, N.; Benenson, I. GIS-based method for assessing city parking patterns. *J. Transp. Geogr.* **2015**, *46*, 220–231. [CrossRef]
19. Yang, B.; Fantini, N.; Jensen, C.S. iPark: Identifying parking spaces from trajectories. In Proceedings of the 16th International Conference on Extending Database Technology, Genoa, Italy, 18–22 March 2013; pp. 705–708.
20. Li, Y.; Luo, J.; Chow, C.-Y.; Chan, K.-L.; Ding, Y.; Zhang, F. Growing the charging station network for electric vehicles with trajectory data analytics. In Proceedings of the 2015 IEEE 31st International Conference on Data Engineering, Seoul, Korea, 13–17 April 2015; pp. 1376–1387.

21. Zhang, F.; Yuan, N.J.; Wilkie, D.; Zheng, Y.; Xie, X. Sensing the pulse of urban refueling behavior: A perspective from taxi mobility. *ACM Trans. Intell. Syst. Technol. (TIST)* **2015**, *6*, 1–23. [CrossRef]
22. Dodge, S.; Weibel, R.; Forootan, E. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Comput. Environ. Urban Syst.* **2009**, *33*, 419–434. [CrossRef]
23. Hosseinpoor Milaghardan, A.; Ali Abbaspour, R.; Claramunt, C. A Spatio-Temporal Entropy-based Framework for the Detection of Trajectories Similarity. *Entropy* **2018**, *20*, 490. [CrossRef]
24. Shen, J.; Liu, X.; Chen, M. Discovering spatial and temporal patterns from taxi-based Floating Car Data: A case study from Nanjing. *GISci. Remote Sens.* **2017**, *54*, 617–638. [CrossRef]
25. Song, X.; Zhang, Q.; Sekimoto, Y.; Shibasaki, R. Prediction of human emergency behavior and their mobility following large-scale disaster. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 5–14.
26. Lye, G.X.; Cheng, W.K.; Tan, T.B.; Hung, C.W.; Chen, Y.-L. Creating Personalized Recommendations in a Smart Community by Performing User Trajectory Analysis through Social Internet of Things Deployment. *Sensors* **2020**, *20*, 2098. [CrossRef] [PubMed]
27. Ye, M.; Yin, P.; Lee, W.-C. Location recommendation for location-based social networks. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 458–461.
28. Lian, D.; Ge, Y.; Zhang, F.; Yuan, N.J.; Xie, X.; Zhou, T.; Rui, Y. Scalable content-aware collaborative filtering for location recommendation. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 1122–1135. [CrossRef]
29. Tu, W.; Cao, R.; Yue, Y.; Zhou, B.; Li, Q.; Li, Q. Spatial variations in urban public ridership derived from GPS trajectories and smart card data. *J. Transp. Geogr.* **2018**, *69*, 45–57. [CrossRef]
30. Yuan, N.J.; Zheng, Y.; Zhang, L.; Xie, X. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Trans. Knowl. Data Eng.* **2012**, *25*, 2390–2403. [CrossRef]
31. Ding, L.; Fan, H.; Meng, L. Understanding taxi driving behaviors from movement data. In *AGILE 2015*; Springer: Washington, WA, USA, 3–7 August 2015; pp. 219–234.
32. Faloutsos, C.; Ranganathan, M.; Manolopoulos, Y. Fast Subsequence Matching in Time-Series Databases. *ACM Sigmod Rec.* **1994**, *23*, 419–429. [CrossRef]
33. Bian, W.; Cui, G.; Wang, X. A Trajectory Collaboration Based Map Matching Approach for Low-Sampling-Rate GPS Trajectories. *Sensors* **2020**, *20*, 2057. [CrossRef]
34. Ta, N.; Li, G.; Xie, Y.; Li, C.; Hao, S.; Feng, J. Signature-based trajectory similarity join. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 870–883. [CrossRef]
35. Khan, R.; Ali, I.; Altowaijri, S.M.; Zakarya, M.; Ur Rahman, A.; Ahmedy, I.; Khan, A.; Gani, A. LCSS-based algorithm for computing multivariate data set similarity: A case study of real-time WSN data. *Sensors* **2019**, *19*, 166. [CrossRef]
36. Papadias, D.; Zhang, J.; Mamoulis, N.; Tao, Y. Query processing in spatial network databases. In Proceedings of the 29th International Conference on Very Large Data Bases-Volume 29, Berlin, Germany, 9–12 September 2003; pp. 802–813.
37. Guan, B.; Liu, L.; Chen, J. Using relative distance and hausdorff distance to mine trajectory clusters. *Indones. J. Electr. Eng. Comput. Sci.* **2013**, *11*, 115–122. [CrossRef]
38. Yi, B.-K.; Jagadish, H.; Faloutsos, C. Efficient retrieval of similar time sequences under time warping. In Proceedings of the 14th International Conference on Data Engineering, Orlando, FL, USA, 23–27 February 1998; pp. 201–208.
39. Tiakas, E.; Papadopoulos, A.; Nanopoulos, A.; Manolopoulos, Y.; Stojanovic, D.; Djordjevic-Kajan, S. Searching for similar trajectories in spatial networks. *J. Syst. Softw.* **2009**, *82*, 772–788. [CrossRef]
40. Chang, J.-W.; Bista, R.; Kim, Y.-C.; Kim, Y.-K. Spatio-temporal similarity measure algorithm for moving objects on spatial networks. In Proceedings of the International Conference on Computational Science and Its Applications, Kuala Lumpur, Malaysia, 26–29 August 2007; pp. 1165–1178.
41. Xia, Y.; Wang, G.-Y.; Zhang, X.; Kim, G.-B.; Bae, H.-Y. Research of spatio-temporal similarity measure on network constrained trajectory data. In Proceedings of the International Conference on Rough Sets and Knowledge Technology, Beijing, China, 15–17 October 2010; pp. 491–498.
42. Abraham, S.; Lal, P.S. Spatio-temporal similarity of network-constrained moving object trajectories using sequence alignment of travel locations. *Transp. Res. Part C Emerg. Technol.* **2012**, *23*, 109–123. [CrossRef]

43. Yuan, Y.; Raubal, M. Measuring similarity of mobile phone user trajectories–a Spatio-temporal Edit Distance method. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 496–520. [CrossRef]
44. Zheng, Y. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol. (TIST)* **2015**, *6*, 29. [CrossRef]
45. Wang, H. The Relationship of Road Network and Urban Efficiency Based on Scaling Law. Master's Thesis, Tsinghua University, Beijing, China, 2015.
46. Iacono, M.; Levinson, D. Mutual causality in road network growth and economic development. *Transp. Policy* **2016**, *45*, 209–217. [CrossRef]
47. Sreelekha, M.; Krishnamurthy, K.; Anjaneyulu, M. Interaction between road network connectivity and spatial pattern. *Procedia Technol.* **2016**, *24*, 131–139. [CrossRef]
48. Cai, X.; Wu, Z.; Cheng, J. Using kernel density estimation to assess the spatial pattern of road density and its impact on landscape fragmentation. *Int. J. Geogr. Inf. Sci.* **2013**, *27*, 222–230. [CrossRef]
49. Burgess, E. *The Growth of the City*; Park, R.E., Burgess, E.W., McKenzie, R.D., Eds.; University of Chicago Press: Chicago, IL, USA, 1925.
50. Hoyt, H. *The Structure and Growth of Residential neighborhoods in American Cities*; US Government Printing Office: Washington, DC, USA, 1939.
51. Harris, C.D.; Ullman, E.L. The nature of cities. *Ann. Am. Acad. Political Soc. Sci.* **1945**, *242*, 7–17. [CrossRef]
52. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; CRC Press: London, UK; Boca Raton, FL, USA, 1986; Volume 26.
53. Miyagawa, M. Spacing of intersections in hierarchical road networks. *J. Oper. Res. Soc. Jpn.* **2018**, *61*, 272–280. [CrossRef]
54. Watanabe, T.; Yamaguchi, T.; Koda, S.; Minatani, K. Tactile map automated creation system using openstreetmap. In Proceedings of the International Conference on Computers for Handicapped Persons, Paris, France, 9–11 July 2014; pp. 42–49.
55. Fan, H.; Zipf, A.; Fu, Q.; Neis, P. Quality assessment for building footprints data on OpenStreetMap. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 700–719. [CrossRef]
56. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the KDD, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
57. Salvador, S.; Chan, P. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.* **2007**, *11*, 561–580. [CrossRef]
58. Keogh, E.; Ratanamahatana, C.A. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* **2005**, *7*, 358–386. [CrossRef]
59. Wang, X.; Mueen, A.; Ding, H.; Trajcevski, G.; Scheuermann, P.; Keogh, E. Experimental comparison of representation methods and distance measures for time series data. *Data Min. Knowl. Discov.* **2013**, *26*, 275–309. [CrossRef]
60. Ontañón, S. An overview of distance and similarity functions for structured data. *Artif. Intell. Rev.* **2020**, 1–43.
61. Tversky, A. Features of similarity. *Psychol. Rev.* **1977**, *84*, 327. [CrossRef]