

Article

Hierarchical Clustering Algorithm for Multi-Camera Vehicle Trajectories Based on Spatio-Temporal Grouping under Intelligent Transportation and Smart City

Wei Wang ¹, Yujia Xie ^{1,*} and Luliang Tang ² 

¹ College of Information Engineering, Nanjing University of Finance & Economics, Nanjing 210023, China; 1120201134@stu.nufe.edu.cn

² State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China

* Correspondence: 9120181003@nufe.edu.cn

Abstract: With the emergence of intelligent transportation and smart city system, the issue of how to perform an efficient and reasonable clustering analysis of the mass vehicle trajectories on multi-camera monitoring videos through computer vision has become a significant area of research. The traditional trajectory clustering algorithm does not consider camera position and field of view and neglects the hierarchical relation of the video object motion between the camera and the scenario, leading to poor multi-camera video object trajectory clustering. To address this challenge, this paper proposed a hierarchical clustering algorithm for multi-camera vehicle trajectories based on spatio-temporal grouping. First, we supervised clustered vehicle trajectories in the camera group according to the optimal point correspondence rule for unequal-length trajectories. Then, we extracted the starting and ending points of the video object under each group, hierarchized the trajectory according to the number of cross-camera groups, and supervised clustered the subsegment sets of different hierarchies. This method takes into account the spatial relationship between the camera and video scenario, which is not considered by traditional algorithms. The effectiveness of this approach has been proved through experiments comparing silhouette coefficient and CPU time.



Citation: Wang, W.; Xie, Y.; Tang, L. Hierarchical Clustering Algorithm for Multi-Camera Vehicle Trajectories Based on Spatio-Temporal Grouping under Intelligent Transportation and Smart City. *Sensors* **2023**, *23*, 6909.

<https://doi.org/10.3390/s23156909>

Academic Editor: Kit Yan Chan

Received: 19 June 2023

Revised: 28 July 2023

Accepted: 1 August 2023

Published: 3 August 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

Vehicle trajectory data analysis is a research hotspot in intelligent transportation and smart city [1]. People can deeply understand their life trajectory, social behavior, environmental change, and urban evolution using video GIS, video object recognition, and visual analysis of video object trajectory [2,3]. In recent years, the urban video monitoring system has gradually developed from single-camera processing to multi-camera equipment joint analysis. The system generates a large number of trajectory data under the multi-camera joint monitoring system and records the movement of people, vehicles, and animals in various scenarios [4], with such characteristics as easy deployment, intuitive information, and rich media expression. Dividing each vehicle trajectory into a suitable cluster by measuring its similarity is a crucial and challenging aspect of trajectory analysis [5]. However, there are some areas for improvement in current clustering methods for video object trajectory research. Specifically, the analysis object is limited to image trajectory [6], ignoring the actual trajectory of the video object in geographical space. Additionally, the traditional trajectory clustering algorithm does not consider the geographic space relation of multi-cameras and only clusters datasets in a single camera and a single scenario, resulting in poor real-time performance and accuracy in large-scale cross-camera or even cross-camera group trajectory data.

Clustering large-scale trajectory data is challenging. First, the trajectory is complex, so the computational cost for clustering analysis is high. Second, the trajectories sampled by the multi-camera joint monitoring system are not equidistant, making it difficult to unify the dimensions between different video object trajectories, so reasonable point correspondence rules should be given to measure the difference in distance calculation. Finally, there is a vision-blind area and a long distance between camera groups, so the direct calculation without simplifying trajectories causes the local differences between trajectories to be overlooked due to the large global scope. The traditional algorithms only cluster the equidistant trajectories of sampling points under a single camera for small-scale datasets without considering the hierarchical relation between camera position, the field of view, and the motion of video objects between cameras and scenarios and without reasonable analysis of cross-group trajectory data. Therefore, this paper proposed a hierarchical clustering algorithm for multi-camera vehicle trajectories based on spatio-temporal grouping. Based on our previous work, we have done further research in this paper [7]. In this paper, the camera groups are divided according to the spatial distribution law of camera equipment in reality, which effectively makes up for the deficiency of traditional methods that do not take into account the spatial relationship between camera equipment. The point correspondence rule of optimal unequal length trajectories and the overlapping scale factor of trajectory distance under camera-joint system are proposed to effectively calculate the distance between unequal multi-camera trajectories and effectively reduce the computational cost compared with the traditional methods. In addition, this paper also proposes a method to automatically obtain labels of video object trajectories in the camera group and across camera groups, which can achieve better clustering results through additional useful supervision information.

This paper is mainly structured as follows: Section 2 introduces the relevant work in the research field involved; Section 3 presents the method description, including the spatialization of video object trajectory, scenarios and division rule of camera groups, point correspondence rule of optimal unequal length trajectory, overlapping scale factor of trajectory distance under camera-joint system, and the details of the supervised video object clustering algorithm within and between camera groups; Section 4 demonstrates the visualization effect of the algorithm on the dataset and evaluates the accuracy and efficiency of the method through experiments; Section 5 discusses the implications of this work, as well as the limitations of this work and how it will be addressed in future works; Section 6 presents the conclusions and prospects.

2. Related Work

The research fields involved in this paper include video-geographic scenario data fusion organization and trajectory clustering. The research status of relevant fields is as follows.

2.1. Video-Geographic Scenario Data Fusion Organization

The data fusion organization of video-geographic scenarios is the basis of video object analysis combining geospatial information. Based on the concepts of Multi Media GIS [8], Geo Video [9], and Video GIS [10], the metadata description method [11] and GPS correlation method [12] were constructed in the earlier research, and geographic retrieval and play of video images were realized by describing the geographic location of video frames.

In recent years, research has focused more on the fusion of video content and geographic scenarios [13]. The fusion of video content and geographic scene can be subdivided into the method of strengthening video through geographic scene and the method of strengthening geographic scene by video [14]. The output result of the former is enhanced video, and the video content is visually enhanced by using the spatial information in the geographic scene [15], while the latter is constructed and presented through the geographic scene [16]. According to the different mapping objects, it can be divided into video frame

picture projection and video object projection. The video frame picture projection maps the video frame image to the geographic scene according to the camera parameters. The video object projection realizes the separation of the front and rear scenes through the related theory of computer vision and projects into the geographical scene. The picture projection of video frame includes the methods of correlation [17], fixed plane play [18], global mapping [19], and so on. The video object projection can be divided into the methods of foreground and background independent projection [20], foreground projection [21], and foreground abstraction [22].

2.2. Trajectory Clustering

Common in pattern recognition, data analysis, and machine learning, trajectory clustering is an efficient method for analyzing trajectory data, aiming to obtain space, time, and even potential information in trajectory data. It is widely applied in such fields as object motion prediction [23], traffic monitoring [24], behavior understanding [25], anomaly detection [26], and 3D reconstruction [27]. In addition, data representation, feature extraction, and distance measurement selection are the key preliminary work of trajectory clustering. For example, a trajectory can be represented as a vector and downsampled to a uniform length, so the Euclidean distance [28] can be used. Trajectories can also be considered as samples of the probability distribution, so Bhattacharyya distance [29] can be used to measure the distance between two trajectories.

According to the availability of labeled data, there are three trajectory clustering methods: unsupervised, supervised, and semi-supervised models. The unsupervised model aims to cluster the data without human experts supervising or labeling the data and obtain the reasoning function by analyzing the unlabeled dataset [30]. The supervised model is learned before trajectory clustering. The labeled data are usually used to learn the function that maps data to its label, the function that predicts unlabeled data clustering [31]. The labeled data often require heavy manual work by experts. The semi-supervised algorithm is trained by labeled data and adjusted by unlabeled data [32].

The unsupervised trajectory clustering algorithm does not rely on other prior knowledge but uses constructors to describe the implicit relationship between trajectory samples. The representative methods are densely clustering models [33], hierarchical clustering models [34], and spectral clustering models [35]. By directly defining the density-reachable rule, densely clustering models can obtain the maximum set of density connected samples as cluster clusters. Hierarchical clustering models are divided into “top-down [36]” type and “bottom-up [37]” type. The former regards each trajectory sample as a cluster and gradually merges into larger clusters by defining the similarity between clusters; the latter gradually divides the trajectory data set into smaller groups by using the idea of divide and conquer until it meets the requirements of clustering. Supervised trajectory clustering algorithm is represented by the K-NN algorithm [38] and statistical model method [39]. The K-NN algorithm uses the average nearest distance as the evaluation criterion to realize trajectory clustering. The statistical model method uses the Gaussian mixture model [40] and other statistical models to construct the trajectory category probability function to obtain the trajectory sample category. In addition to the above methods, in recent years, with the continuous extension of deep learning, there are a large number of deep learning algorithms [41,42]. Semi-supervised trajectory clustering algorithm [43,44] fully combines supervised clustering algorithm with unsupervised clustering algorithm to reduce the label labor cost and minimize the over-fitting problem in the clustering process. The basic idea is to update the classifier by classifying the trajectory data and clustering the new data.

In order to understand the three kinds of algorithms more clearly, we give the tabular overview of different trajectory clustering algorithms in Table 1.

Table 1. Tabular overview of different trajectory clustering algorithms.

Category	Classification	Time Complexity	Antinoise Ability	Labor Cost	Measurement	Representative Algorithm
Unsupervised	Densely Models	$O(n^*log n)$	General	low	Density	DBCSAN [33]
	Hierarchical Models	$O(n^2 log n)$	Strong	low	Distance	HITS [36]
	Spectral Models	$O(n^3)$	Strong	low	Distance	CURE [37]
Supervised	Nearest Neighbor	$O(n)$	weak	high	Distance	K-NN [39]
	Statistical Models	$O(n^2)$	weak	high	Pattern mining	GMM [41]
	Neural Network	Depends on network	General	high	Deep network characteristics	CNN [42]
Semi-supervised	Invented from unsupervised or supervised algorithms	Related to the invented algorithm	Related to the invented algorithm	General	Related to the invented algorithm	Modified Hierarchical Clustering Models [43], modified Statistical Models [44], etc.

3. Method Description

3.1. Spatialization of Video Object Trajectory

Building the mapping relation between each camera image square plane and geographical space object square plane is necessary to transform from image space coordinates to geographical space coordinates. This paper used the contact point between the video object subgraph and the ground as the locating point and sampled at a certain time interval [45]. The video object trajectory in the geographic scenario was obtained from a building mapping relation between image space and geographic space [46], as shown in Figure 1.

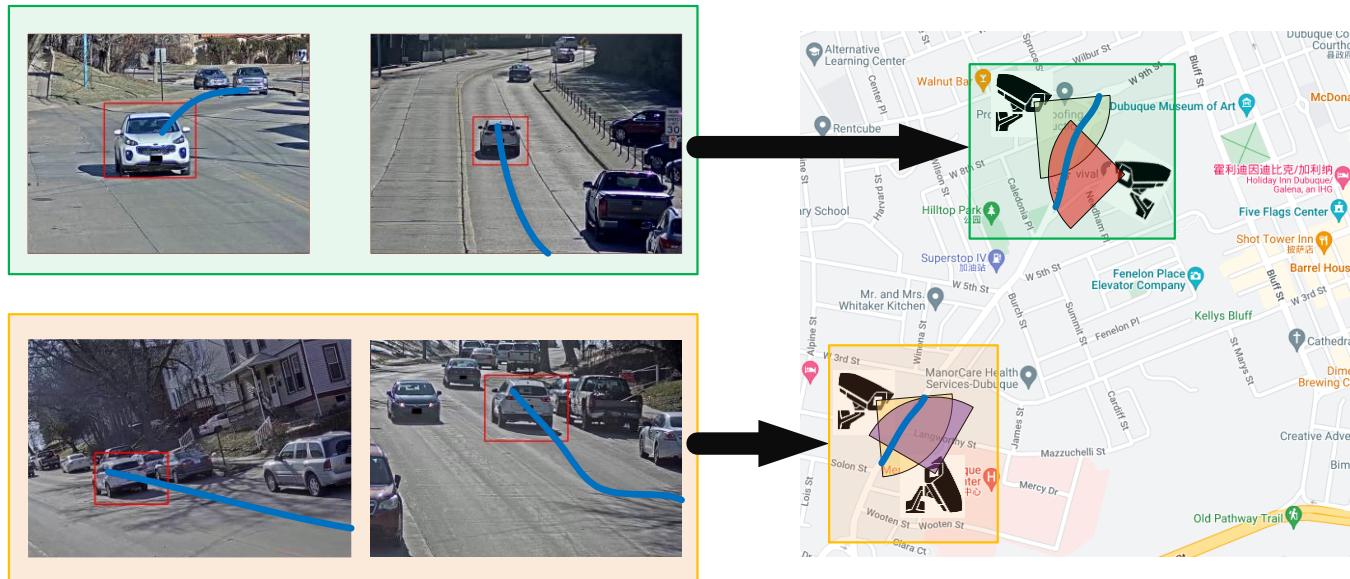


Figure 1. Schematic diagram of video object trajectory spatialization (blue lines represent the image space trajectories and the geospatial trajectories of the vehicle video object).

This study uses the homography matrix method to construct a mapping model. If the image coordinate of a point is q , and the geographic space coordinate is Q , then the homogeneous coordinate of q and Q can be represented as

$$q = [x \ y \ 1]^T, \quad (1)$$

$$Q = [X \ Y \ Z \ 1]^T. \quad (2)$$

If the mapping matrix is M , the relational expression between q and Q is

$$q = MQ. \quad (3)$$

The image plane is scaled, translated, and rotated to a geospatial plane, so the mapping matrix M can be decomposed into

$$M = s \cdot W \cdot R, \quad (4)$$

where s is the scale factor; W is the camera translation transformation matrix; R is a 3×4 dimensional rotation transformation matrix.

$$W = \begin{bmatrix} f_u & 0 & u \\ 0 & f_v & v \\ 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$R = [r_1 \ r_2 \ r_3 \ e] \quad (6)$$

where f_u and f_v represent the product of the physical focal length of the lens and the sensor size in the horizontal and vertical axis directions of each unit, respectively; u and v represent the offset of the imaging center relative to the principal optic axis on the horizontal and vertical axis, respectively; r_1 , r_2 , and r_3 represent the rotation relation of the coordinate system in the X-axis, Y-axis, and Z-axis in the physical space, respectively; e is the translation relation between coordinate systems.

When using the homography matrix method, it is assumed that the camera field of view plane in geographic space is horizontal—that is, $Z = 0$ at the plane. Thus, the mapping relation between image and geographic space can be regarded as the mapping from one plane to another. To simplify the calculation, the Z in Q and the r_3 rotated about the Z-axis in R are removed. Hence, the homography matrix M is simplified as

$$M = s \cdot \begin{bmatrix} f_u & 0 & u \\ 0 & f_v & v \\ 0 & 0 & 1 \end{bmatrix} \cdot [r_1 \ r_2 \ r_3 \ e]. \quad (7)$$

The geospatial coordinates of the video object trajectory can be obtained according to the solution of the matrix M .

3.2. Scenarios and Division Rule of Camera Groups

The closely adjacent location cameras within the group and the overlapped camera field of view are preferred when cameras are divided into groups. The weigh options rules between the two can be specified as

$$Sel_{rule} = \frac{1}{\varepsilon} \underset{cam \in C}{argmax} P(D|h)P(h), \quad (8)$$

where C represents all cameras to be divided, and $P(h)$ represents the nearest neighbor in a position, which can be stated as the reciprocal of the distance between positions:

$$P(h) = \frac{1}{\sum \|loc(cam_u) - loc(cam_v)\|}. \quad (9)$$

$P(D|h)$ is the degree of view overlap, expressed by the area of view overlap:

$$P(D|h) = \sum area(cam_u) \cup area(cam_v). \quad (10)$$

ε represents the normalized constant parameter.

3.3. Point Correspondence Rule of Optimal Unequal Length Trajectories

Inspired by the dynamic time warping [47], to better compare the distances between two trajectories with different lengths, it is necessary to establish one-to-many and many-

to-one matching so that the two trajectories have the same pattern of wave troughs and peaks perfectly matched. Two unequal-length trajectory sequences are known:

$$Tra_x = \{p_{x,j} (j = 1, \dots, len_x)\}, Tra_y = \{p_{y,j} (j = 1, \dots, len_y)\}, len_x \neq len_y. \quad (11)$$

The trajectory sequence may not have equidistant time points. A feature space represented by \mathcal{F} is fixed. Its local distance measure is defined as a function:

$$c : \mathcal{F} \times \mathcal{F} \rightarrow \mathcal{R} \geq 0. \quad (12)$$

The defined sequence is the point correspondence sequence of optimal unequal-length trajectory:

$$Seq(S) = (s_1, s_2, \dots, s_e, \dots, s_{len(Seq(S))}). \quad (13)$$

$$s_e = (n_e, m_e) \in [1 : N] \times [1 : M] \quad (14)$$

Then, the sequence $Seq(S)$ satisfies

$$\begin{cases} s_1 = (1, 1) \\ s_{len(Seq(S))} = (N, M) \\ n_1 \leq n_2 \leq \dots \leq n_{len(Seq(S))}, m_1 \leq m_2 \leq \dots \leq m_{len(Seq(S))} \\ s_{e+1} - s_e \in \{(1, 0), (0, 1), (1, 1)\} \end{cases}. \quad (15)$$

The distance cost between Tra_x and Tra_y is written as

$$c_p(Tra_x, Tra_y) = \sum_{l=1}^L c(Tra_{x,n_l}, Tra_{y,m_l}). \quad (16)$$

The optimal total cost between Tra_x and Tra_y is shown as the minimum value of $c_p(Tra_x, Tra_y)$:

$$\min\{c_p(Tra_x, Tra_y)\}. \quad (17)$$

3.4. Overlapping Scale Factor of Trajectory Distance under Camera-Joint System

This part explains the increase in distance calculation due to the time overlap of multiple cameras in the camera-joint system. Then it gives a method to obtain the scale factor of trajectory distance overlap to offset the increase in distance.

Assuming G as a camera group,

$$G = \{Cam_1, Cam_2, Cam_3, \dots, Cam_n\}. \quad (18)$$

For \forall video object $Obj_i \in G$, it is assumed that Obj_i is captured by $G_i = \{Cam^{(1)}, Cam^{(2)}, \dots, Cam^{(s)}, \dots, Cam^{(t)}\}$, where

$$G_i \subseteq G. \quad (19)$$

According to the capture sequence, the trajectory corresponding to the video object can be expressed as a series of nodes:

$$Tra_i = \{p_{i,j} (j = 1, \dots, len_i)\}, \quad (20)$$

$$p_{i,j} = (Cam_{i,j}, x_{i,j}, y_{i,j}, t_{i,j}), \quad (21)$$

where

$$Cam_{i,j} \in G_i, \quad (22)$$

which represents that the video object is captured by $Cam_{i,j}$ at this node.

Assuming the two trajectories Tra_1 and Tra_2 under the camera group formed by cameras Cam_1 and Cam_2 are shown in Figure 2.

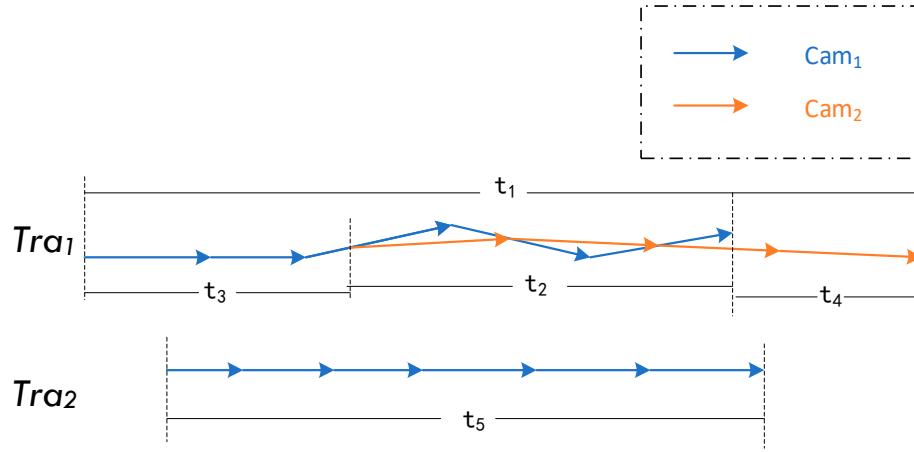


Figure 2. Schematic diagram of trajectory capture camera overlay.

In Tra_1 , the total duration under two cameras is t_1 , which can be divided into t_2 , t_3 , t_4 , in which t_3 , t_4 are captured by only one camera while t_2 is captured by two cameras at the same time; Tra_1 is only captured by Cam_1 during time t_5 . If the distance between the two is calculated directly, the density of trajectory nodes will increase as cameras overlap in time t_2 of Tra_1 , leading to an increase in the distance value. Therefore, Tra_1 is obtained first to eliminate the distance of the added part. The total global duration of Tra_2 is $\alpha_1 = t_1$, $\alpha_2 = t_5$, respectively, while the actual calculation duration is $\beta_1 = t_1 + t_2$, $\beta_2 = t_5$. Therefore, the scale factor of overlapping trajectories of Tra_1 and Tra_2 can be obtained as follows:

$$k_i = \frac{\alpha_{i1} + \alpha_{i2}}{\beta_{i1} + \beta_{i2}}. \quad (23)$$

This factor should be multiplied in the distance calculation to eliminate the increase in distance due to camera overlap.

This part details the trajectory clustering algorithm within the camera group (SCAIMG) and between camera groups (SCAIBG). The process of the algorithm is shown in Figure 3.

3.4.1. Supervised Clustering Label Acquisition

Supervised clustering results can be obtained with more additional useful supervised information than unsupervised clustering. For the video object trajectory under the camera joint system in this chapter, the “main camera” of video objects is defined as the label for the supervised clustering of trajectory samples. For the trajectory between camera groups, the number sequence of the camera group passed through is used as the label of the supervised clustering of the trajectory sample.

In the camera group of the camera-joint system, the camera number with the most capture times should be the same among video objects with the same characteristics. Therefore, for accurate trajectory data clustering, we use the camera number with the highest capture times corresponding to each video object as the label of supervised clustering:

$$label(Tra_i) = \max_{Cam_{i,j} \text{ correspond to } Tra_i} NUM(Cam_{i,j}), \quad (24)$$

Furthermore, we used the camera group number as the label of supervised clustering between camera groups.

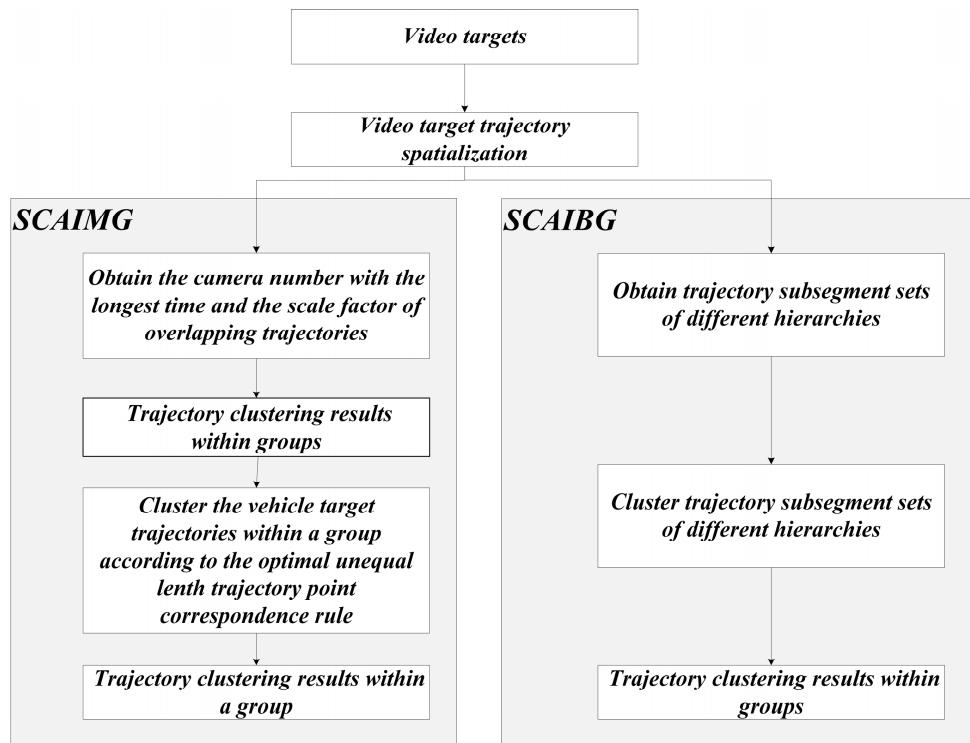


Figure 3. Flow chart of hierarchical clustering algorithm for vehicle trajectory considering camera information under the multi-camera joint monitoring system.

3.4.2. Trajectory Clustering Algorithm within the Camera Group

The trajectory clustering algorithm within the camera group aims to obtain a group of clustering centers. First, the video objects in the camera group are spatialized, followed by clustering their trajectory. The pseudo-code of trajectory clustering within the group (SCAIMG) is shown in Algorithm 1.

Algorithm 1: Supervised trajectory clustering algorithm considering camera information in the multi-camera collaborative monitoring group (SCAIMG)

```

1 : Input : A camera group  $G = \{Cam_1, Cam_2, Cam_3, \dots, Cam_n\}$ , the trajectory set :  $T_G = \{Tra_i\}$ ,  $Tra_i = \{p_{i,j} | j = 1, \dots, len_i\}\}$ .
2 : Output : a group of trajectory clustering centers  $C = \{c_1, c_2, \dots, c_\lambda\}$ .
3 : Obtain the capture time of each video object under each camera according to Section 4.1 and use the camera number with the most capture times as the label :  $label(Tra_i) = \max_{Cam_{i,j} \in Tra_i} NUM(Cam_{i,j})$ . Initialize the new set  $\phi = T_G$ , set clusters number  $K$ , initialize set  $all\_vectors\_updated = [0, 0, \dots, 0]$  (size =  $K$ ); Randomly select  $K$  samples from  $\Phi$  as initial cluster centers  $C \leftarrow \{c_1, c_2, \dots, c_\lambda\}$ ; Set the number of iterations parameter  $epochs$ ; Set the learning rate  $\eta \in (0, 1)$ .
4 : According to Equation (23), obtain the overlap scale factor of each trajectory :  $k_i \leftarrow \frac{\alpha_{i1} + \alpha_{i2}}{\beta_{i1} + \beta_{i2}}$ .
5 :  $step = 0$ 
6 : While not each vector in  $C$  is updated and  $step < epochs$ :
7 :   For each sample  $s$  from  $\phi$ :
8 :     calculate the distance between  $s$  to each center  $c_\tau$ :  $d(s, c_\tau) \leftarrow \min\{c_p(s, c_\tau)\}$ 
9 :      $c_{\tau^*} \leftarrow c_\tau$ , where  $\tau^* = \arg \min_{\tau \in \{1, 2, \dots, K\}} d(s, c_\tau)$ 
10 :    If  $label(s) == label(c_{\tau^*})$  :
11 :       $s' \leftarrow s / (1 - \eta) \cdot d(s, c_{\tau^*})$ 
12 :    Else:
13 :       $s' \leftarrow s / (1 + \eta) \cdot d(s, c_{\tau^*})$ 
14 :     $all\_vectors\_updated[s] = 1$ 
15 : Return  $\phi$ 
  
```

In Algorithm 1, line 11 and line 13 represent the operations of “approaching” and “moving away”, respectively. The schematic diagram of the approaching operation is shown in Figure 4. It is assumed that $label(s) == label(c_\tau)$ is known, and every node in the trajectory s moves towards c_τ .

The trajectory of the “moving away” operation moves in the opposite direction in Figure 4.

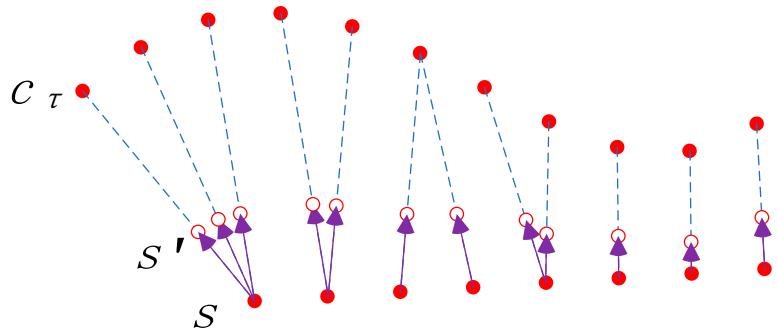


Figure 4. Schematic diagram of trajectory “approaching” the cluster center.

3.4.3. Trajectory Clustering Algorithm between Groups

We can combine the entry and departure points in each camera field of view of video object trajectories to reflect the dynamic characteristics of group trajectories [48,49] because the range of trajectories between groups is larger than that within a group. The entry point and departure point of the vehicle video object under each camera group are taken as the trajectory sampling points to form multiple trajectory subsegments, which are then hierarchically clustered.

The sampled trajectory is supposed as:

$$f_i = \left\{ p_{i,j}, (j = 1, \dots, \text{len}_{f_i}) \right\}, \quad (25)$$

$$p_{i,j} = (Group_{i,j}, x_{i,j}, y_{i,j}, t_{i,j}), \quad (26)$$

where $p_{i,j}$ represents the entry point or departure point information, and $Group_{i,j}$ represents the j -th camera group number that video f_i passes through. $x_{i,j}, y_{i,j}$ represent the coordinates, and $t_{i,j}$ represents the timestamp.

Figure 5 shows the trajectory f_1, f_2, f_3 in camera Groups 1–3. f_1 passes through Group 1, Group 2, and Group 3, with five hierarchies; f_2 passes through Group 1 and Group 2, with three hierarchies; f_3 passes through Group 3, with one hierarchy. f_1, f_2 contain trajectory subsegments of three hierarchies starting at Group 1 and ending at Group 2; f_1, f_3 contain trajectory subsegments in Group 3, with one hierarchy.

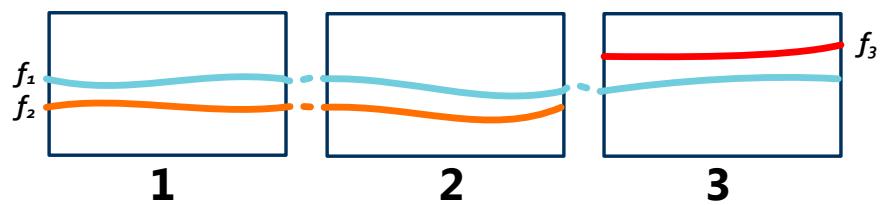


Figure 5. Trajectory subsegments.

We proposed a trajectory clustering algorithm between camera groups with multi-hierarchies to reasonably analyze the cross-camera group trajectories. The algorithm aims to obtain the λ sets of cluster centers:

$$\{C(1), C(3), \dots, C(2h-1), \dots, C(\lambda)\}, h \in \mathbb{N}, \quad (27)$$

where

$$C(2h-1) = \{c_{2h-1,1}, c_{2h-1,2}, \dots, c_{2h-1,k}, \dots, c_{2h-1,l}\}, h, k \in \mathbb{N}, \quad (28)$$

$$c_{2h-1,k} = \text{average} \left(\{f_{2h-1,m}, f_{2h-1,m+1}, \dots, f_{2h-1,m+2(\lambda-1)}\} \right), \quad (29)$$

$$\{f_{2h-1,m}, f_{2h-1,m+1}, \dots, f_{2h-1,m+2(\lambda-1)}\} \in c_{2h-1,k}, \quad (30)$$

where λ refers to the number of cross-camera groups of trajectories within a group, $C(2h - 1)$ refers to the cluster center set when the number of cross-camera groups of trajectories within groups is $2h - 1$, and k refers to the k -th cluster center.

The pseudo-code of Supervised trajectory clustering algorithm considering camera information between groups (SCAIBG) is shown in Algorithm 2.

Algorithm 2: Supervised trajectory clustering algorithm considering camera information between groups (SCAIBG)

```

1 : Input : a group cross-camera groups video object trajectories :  $f_i = \{p_{i,j}, (j = 1, \dots, len_{f_i})\}$ 
2 : Output :  $\lambda$  groups of trajectory clustering centers  $\{C(1), C(2), \dots, C(\lambda)\}$ .
3 : For  $t \leftarrow 1$  to  $\lambda$  :
4 :   Initialize/newset  $\Phi$ 
5 :   For  $u \leftarrow 1$  to  $obj\_num$ :
6 :     If  $Cnum(f_{obj(u)}) \geq \lambda$  //  $Cnum(f_{obj(u)})$  indicates the total number of camera groups passed by  $obj(u)$ 
7 :       For  $v \leftarrow 0$  to  $u - \lambda - 1$ :
8 :         Add  $[f_u^{(v)}, f_u^{(v+1)}, \dots, f_u^{(v+\lambda)}]$  to  $\phi$  //  $f_u^{(v)}$  represents the  $v$ -th trajectory subsegment of the  $u$ -th
video object
9 :   Set clusters number  $K_\lambda$ 
10 :  Randomly select  $K_\lambda$  samples from  $\phi$  as Initial cluster centers :  $\{C(\lambda)\} = \{c_{t,1}, c_{t,2}, \dots, c_{t,K_\lambda}\}$ 
11 :  Set  $epochs$  as the maximum number of iterations
12 :  Repeat :
13 :    For each sample  $s$  from  $\Phi$ :
14 :       $c_{t,w^*} \leftarrow c_{t,w}$ , where  $w^* = \arg \min_{w \in \{1,2,\dots,K_\lambda\}} d(s, c_{t,w})$ 
15 :      If  $label(s) = label(c_{t,w^*})$ :
16 :         $s' = s / (1 - \eta) \cdot d(s, c_{t,w^*})$ 
17 :      Else
18 :         $s' = s / (1 + \eta) \cdot d(s, c_{t,w^*})$ 
19 :    Until iterations exceed  $epochs$  or there is no change in cluster centers
20 :     $t \leftarrow t + 1$ 
21 :  Return  $\phi$ 

```

4. Experimental Analysis

This part briefly introduces the experimental conditions and data and the trajectory clustering results within and between camera groups. Compared with traditional clustering algorithms, the advantages of the proposed algorithm are demonstrated by silhouette coefficient evaluation, and the time complexity is analyzed to prove the algorithm's effectiveness.

4.1. Experimental Conditions and Data

The experimental data in this paper come from the CityFlowV2 dataset [50] newly launched by NVIDIA, the first large-scale dataset in the world to support cross-camera vehicle tracking, accommodating more than ten intersections and 40 cameras at the same time, with a spatial span greater than 3 km^2 . This dataset contains high-definition synchronous videos collected in Dubuque, the United States, including scenarios such as residential areas and expressways.

The experimental data in this paper are all the original dataset's video sequences. The experimental environment comprises software (Windows10, python 3.6 + sklearn0.0) and hardware (Intel (R) Core (TM) i7-10510U CPU @ 1.80 GHz 2.30 GHz, RAM 12.0 GB, and NVIDIA GeForce MX250). The algorithm to obtain the video object trajectory by preprocessing is as follows: the video dynamic object detection algorithm is Mask-RCNN [51], the tracking algorithm is TNT [52], and the cross-camera re-recognition algorithm is Deep SORT [53].

4.2. Camera Grouping of the CityFlowV2 Dataset

The CityFlowV2 dataset is divided into groups according to Equations (9) and (10). Partial division results are shown in Figure 6. In addition to the groups shown in Figure 6, Cam 1 to Cam 5 are a group, and Cam 6 to Cam 9 are a group.

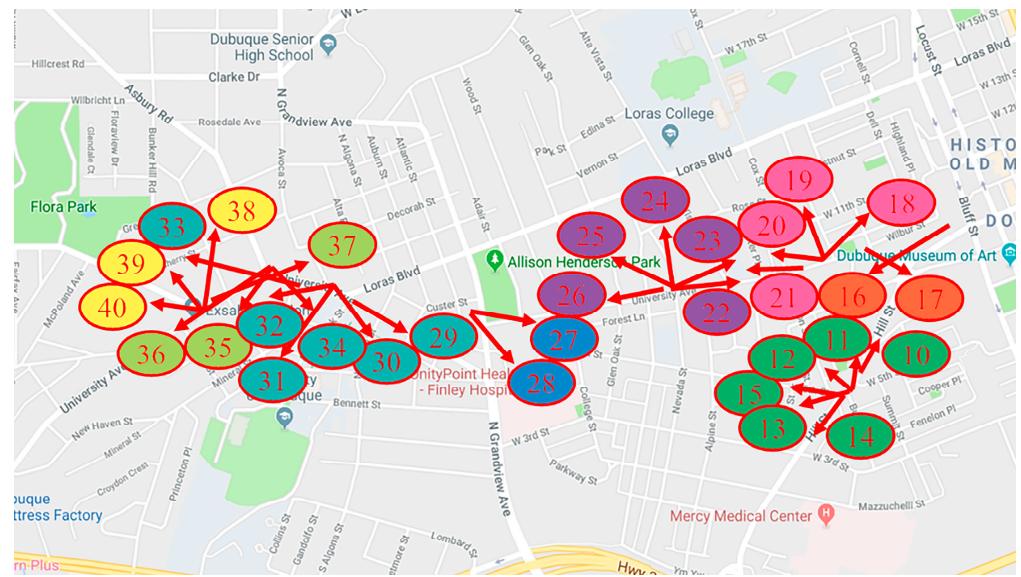


Figure 6. Camera Grouping Results of the CityFlowV2 Dataset (numbers represent camera numbers, red arrows represent the main optical axis direction of the camera lens, and ellipses of different colors represent different camera groups).

4.3. Determination of the Number of Cluster Centers

The number of cluster centers is determined using the elbow method [54]. Due to the continuous optimization of the cluster center and smaller quadratic sum, the first inflection point appearing in the change of the quadratic sum is chosen as the best K value.

$$\min(SSE). \quad (31)$$

$$SSE = \sum_k^{i=1} \sum_{p \in C_i} |p - m_i|^2. \quad (32)$$

For SCAIMG and SCAIBG, the cluster center value of each camera group is determined using the elbow method, as shown in Figures 7 and 8. The number of cluster centers of each camera is the abscissa K corresponding to the blue box.

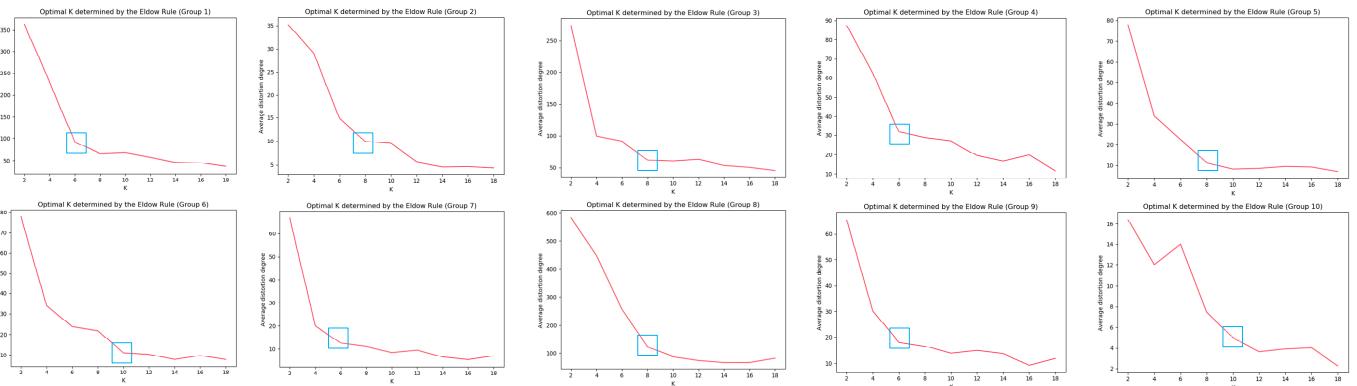


Figure 7. Schematic diagram for determining the number of cluster centers of camera groups in SCAIMG (blue boxes represent the optimal K value).

According to Figures 7 and 8, we can respectively determine the reasonable number of cluster centers K according to the number of cluster centers in each group—the inflection point in the curve of average distortion degree.

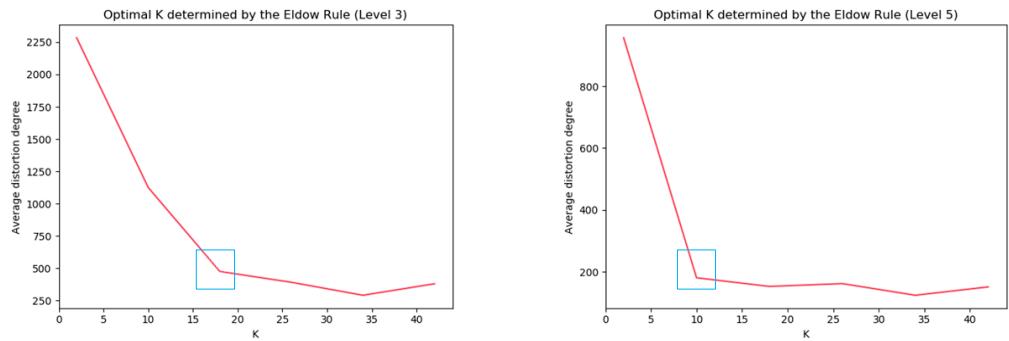


Figure 8. Schematic diagram for determining the number of clustering centers of three and five hierarchies in SCAIBG (blue boxes represents the optimal K value).

4.4. Determination of the Initial Cluster Center

If the initial cluster centers for SCAIMG and SCAIBG are randomly selected, this usually results in slightly different clusters at the end when the clustering algorithm is rerun. In order to solve the above problem, we refer to the work of the other literature [55], make improvements, and propose the initial cluster center acquisition method.

Assume that there are L trajectories for clustering, and the number of supervised clustering label categories is ζ . First, a trajectory sample is selected from each category—that is, a total of ζ trajectory samples are selected as the initial cluster centers, and then the sample with the largest average distance from ζ centers is selected from the remaining $L - \zeta$ samples as the $\zeta + 1$ -th initial cluster center, and so on until K initial cluster centers are selected.

4.5. Algorithm Results and Effect Evaluation

4.5.1. Trajectory Clustering Results within a Group

First, we demonstrate the visualization results of clustering ten camera groups with overlapping cameras in each group. For example, the visualization effect of the trajectories in group 1 is shown in Figure 9:



Figure 9. Schematic diagram of trajectory acquisition of each camera in Group 1 (numbers represent camera numbers, red arrows represent the main optical axis direction of the camera lens, and different color lines represent trajectories under different cameras).

The trajectories under each camera group were obtained in chronological order. The clustering results of SCAIMG in each group are shown in Figure 10. The parameters are set to $epochs = 500, \eta = 0.01$. The red line represents the vehicle trajectory, while the green line is the cluster center.

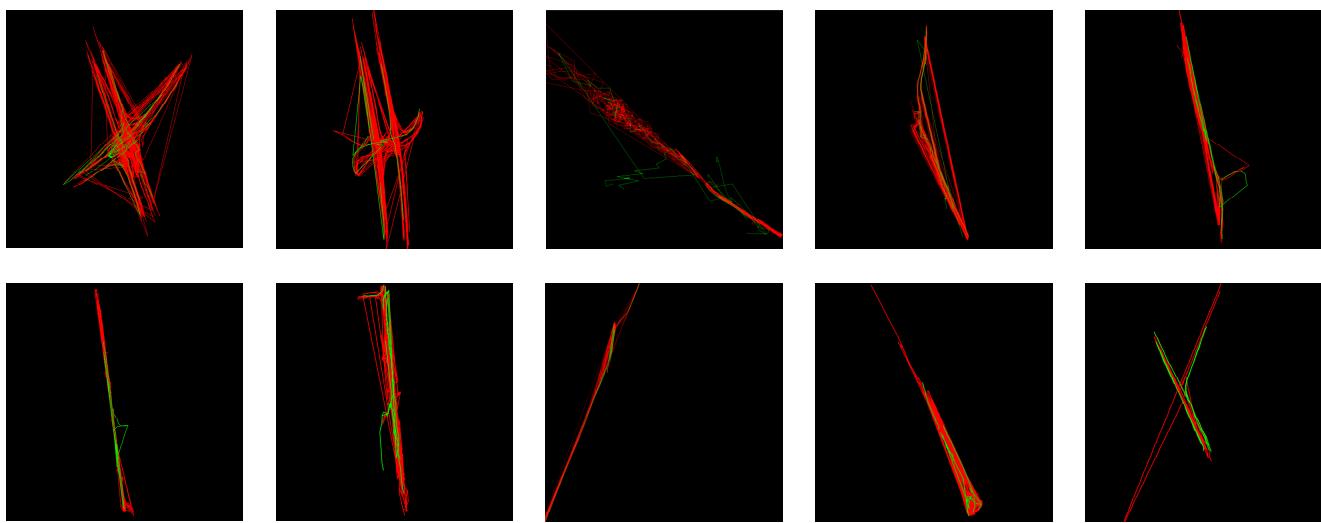


Figure 10. Schematic diagram of 10 groups of vehicle trajectory clustering in the CityflowV2 dataset (red lines represent the trajectory, and green lines represent the cluster center).

Through the visualization effect, a small number of cluster centers represent the overall trend of trajectory data of 10 camera groups.

4.5.2. Trajectory Clustering Results between Camera Groups

In this paper, we performed clustering analysis on trajectory subsegments with three and five hierarchies. There are trajectory subsegments with seven, nine, and eleven or even more hierarchies. However, in these cases, there is a small number of trajectory subsegments, making the clustering analysis lose its practical significance because clustering is to extract a large number of trajectories of the overall trend.

The number of the trajectory subsegments at different hierarchies is shown in Table 2:

Table 2. Number of trajectory subsegments (three and five hierarchies).

Hierarchies	Number of Trajectory Subsegments
3	460
5	391

The results of vehicle trajectory clustering between groups with three hierarchies (upper) and five hierarchies (lower) are shown in Figure 11.

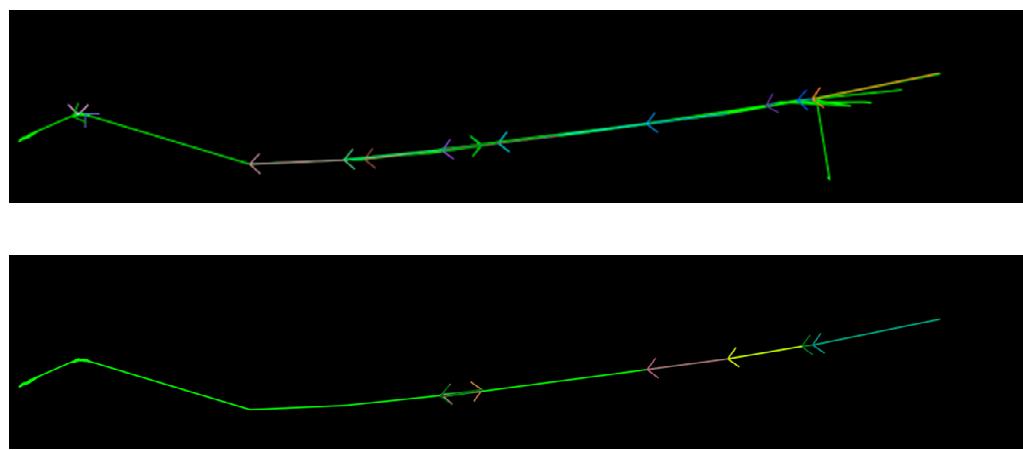


Figure 11. Schematic diagram of vehicle trajectory clustering between groups (three and five hierarchies; arrows of different colors represent cluster centers).

The trajectories between groups have a larger geographical range than those within a group. It can be seen from the visualization effect that the SCAIBG has achieved the overall trend of vehicles moving between groups. The green line represents all vehicle trajectories between groups, while the arrows in other colors represent the overall trend of vehicles moving at different hierarchies.

4.5.3. Cluster Effect Evaluation

The silhouette coefficient [56] is used to compare and analyze the traditional DBSCAN-based method [57] with the proposed algorithm to verify the effectiveness of the proposed method. The average distance between the sample and other samples in the same cluster and the average distance between the sample and the next nearest cluster are combined for evaluation:

$$\text{Silhouette Coefficient}(\zeta) = |1 - a(\zeta)/b(\zeta)| \in [-1, 1]. \quad (33)$$

Figure 12 shows the *Silhouette Coefficient* comparison of trajectory clustering within a group.

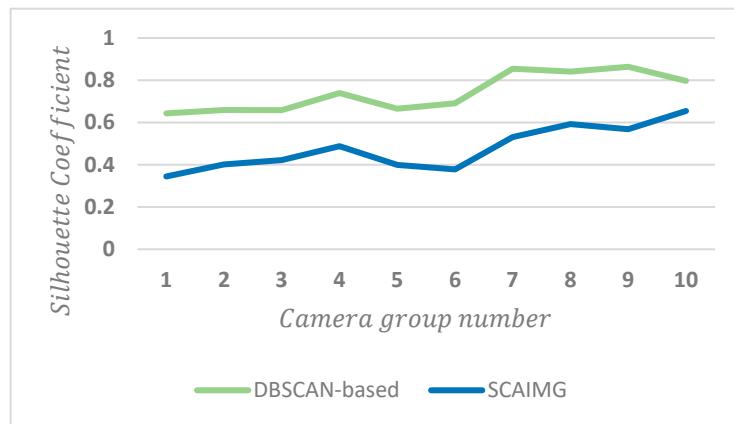


Figure 12. Silhouette coefficient comparison of trajectory clustering within a camera group.

Figure 13 shows the silhouette coefficient comparison of trajectory clustering between groups.



Figure 13. Silhouette coefficient comparison of trajectory clustering between camera groups.

For SCAIMG, the point correspondence of trajectory is considered when trajectory clustering within a group is so that the unequal-length trajectories can move reasonably according to the corresponding relation. The distance overlapping scale factor is used to offset the unreasonable distance increase between trajectories caused by camera overlap to obtain accurate trajectory centers. However, the traditional algorithm does not consider the corresponding relation between trajectories. It can only obtain the approximate distance between trajectories when measuring the trajectory similarity, so the cluster center obtained

from the clustering results can not accurately represent the overall trend of trajectories. In addition, the camera number with the most capture times is used as the clustering label for supervised clustering, considering the spatial relationship between the camera and the video scenario. Figure 12 reveals that the SCAIMG is superior to the traditional trajectory clustering algorithm in clustering each camera group in large-scale vehicle trajectory data such as CityFlowV2.

For the trajectories between groups, similar to SCAIMG, SCAIBG takes group number as the cluster label for supervised clustering and considers the spatial position of camera groups and the spatial relation of video scenario, so the clustering effect is also better than that of the traditional algorithm, as shown in Figure 13.

4.5.4. Algorithm Time Analysis

For SCAIMG, it is assumed that there are n video objects. It takes $O(n)$. to obtain the overlap scale factor for each trajectory. The corresponding relation and distance between each pair of trajectories can be calculated before the iteration. The corresponding relation between each pair of trajectories can be accelerated to $O(n)$ through coarse-graining, projection, and fine-graining [58], so it takes time $\leq n \cdot O(n) = O(n^2)$, in general,

$$T_M(n) \leq O(n^2). \quad (34)$$

Therefore, the algorithm maintains linear time complexity, proving that the algorithm is effective.

Under the hardware given, the CPU time required by SCAIBG on the CityFlowV2 dataset is shown in Table 3.

Table 3. The CPU time required by SCAIMG for each group in the CityFlowV2 dataset.

Hierarchies	Number of Trajectory Subsegments
1	5.693
2	10.165
3	14.770
4	6.558
5	0.388
6	1.343
7	0.359
8	0.559
9	0.465
10	2.642

The algorithm can obtain clustering results in a few seconds or even one second in groups other than groups 2 and 3. The longer time in groups 2 and 3 is because some trajectories within the group are long, taking it extra time to calculate the distance. Overall, SCAIMG can quickly extract the cluster center of large-scale vehicle trajectories.

For the SCAIBG, the clustering results in λ group of prototype vectors, with obj_num video objects. For each video object $obj(u)$, it passes through $Cnum(f_{obj(u)})$ camera groups. Therefore, there should be $u - \lambda - 1$ subsegment of the trajectory under hierarchy u . It takes $O(K_t)$ to randomly select K_t samples. In general, if the iterations in the learning phase reach $epochs$ or the cluster center does not change, it will take a time of

$$T_B(n) \leq O\left(Iter_{max} \cdot \sum_{u=1}^{obj_{num}} Cnum(f_{obj(t)}) \cdot K_t\right) = O(n). \quad (35)$$

$Iter_{max}$, obj_{num} , $Cnum(f_{obj(t)})$, and K_t represent the maximum number of iterations, the number of video objects, the number of cross-camera trajectories corresponding to

video objects, and the number of cluster centers, respectively. The linear time complexity of the algorithm proves that the algorithm is effective.

Under the hardware given, the CPU time required for different hierarchies of trajectories is shown in Table 4.

Table 4. The CPU time required by SCAIBG for each group in the CityFlowV2 dataset.

Hierarchies	Number of Trajectories	CPU Time
3	14.770	0.007
5	0.388	0.036

SCAIBG can effectively simplify the calculation of regular distance by extracting the start and end points of the trajectory. Therefore, it can be seen from Table 4 that the SCAIBG performs quite fast in extracting cluster centers between groups in large-scale vehicle datasets.

5. Discussion

The purpose of “Hierarchical Clustering Algorithm for Multi-camera Vehicle Trajectories” proposed in this paper is to represent vehicle video objects with similar motion patterns by obtaining different levels of video object clustering centers as the representative of vehicle video object motion patterns. The method in this paper has and is not limited to the following practical applications:

- (1) Multi-scale vehicle path prediction. According to prior theories, we can combine time information to analyze the multi-scale prediction of vehicle moving path.
- (2) Urban Planning and Road Planning. Different levels of clustering results, combined with road constraints, provide effective experience and technical support for urban planning departments and traffic management departments.
- (3) Trajectory outliers detection. The clustering results can provide a prerequisite for vehicle trajectory outliers detection. Trajectory outliers are a small number of trajectories that are obviously different from other trajectories data in the trajectory data set, so vehicle trajectory outliers can be analyzed through the distance between the vehicle trajectories and clustering centers.

In the above applications, we think that the trajectory clustering algorithm should ensure high accuracy and efficiency. High accuracy is the premise of the accuracy of the follow-up work, and efficiency is the rigid requirement of some practical applications. For example, path prediction analysis is applied to navigation planning and trajectory anomaly detection to security departments. Rapid cluster analysis enables these applications to be implemented safely.

In this paper, the elbow method is used to determine the optimal number of clusters in clustering analysis, but this method may have some shortcomings in practical application, because the vehicle trajectory has road constraints, and there is a deviation simply according to the elbow method to determine the clustering center without taking into account the actual situation.

Therefore, in the future scientific research work, the final number of clustering centers should be determined by combining the algorithm and road constraints.

6. Conclusion and Prospects

This paper proposed a multi-camera vehicle trajectory hierarchical clustering algorithm based on spatio-temporal grouping, aiming at the problem that the traditional trajectory clustering algorithm did not consider the camera position information, the field of view, and the hierarchical relation of the video object movement between the camera and the scenario. This method used camera number and group number as labels and was a supervised clustering algorithm considering camera information applied to trajec-

tories within and between groups, respectively, enabling trajectories to be classified into reasonable clustering centers.

The experiment showed the clustering results and the comparison and analysis of silhouette coefficients between the proposed method and other clustering methods, and the time complexity analysis proved the proposed algorithm's accuracy, effectiveness, and high efficiency. The limitation of this paper lies in the lack of trajectory clustering analysis combined with spatial semantic features, which will be improved in future research.

Author Contributions: Conceptualization, W.W., Y.X. and L.T.; methodology, W.W.; software, W.W.; formal analysis, W.W., Y.X. and L.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (41801305) and the Open Research Fund of State Key Laboratory of Surveying, Mapping and Remote Sensing Information Engineering, Wuhan University (21S03).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: CityFlowV2 dataset at <https://www.aicitychallenge.org> (accessed on 2 July 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Srivastava, S.; Delp, E.J., III. Video-based real-time surveillance of vehicles. *J. Electron. Imaging* **2013**, *22*, 041103. [[CrossRef](#)]
2. Zhang, G.; Jia, S.; Zhang, X.; Li, X. Saliency-based foreground trajectory extraction using multiscale hybrid masks for action recognition. *J. Electron. Imaging* **2018**, *27*, 053049. [[CrossRef](#)]
3. Yan, X.; Yang, J.; Liu, Y.; Song, L. Multimodal based attention-pyramid for predicting pedestrian trajectory. *J. Electron. Imaging* **2022**, *31*, 053008. [[CrossRef](#)]
4. Wang, Z.; Yuan, X. Visual analysis of trajectory data. *J. Comput.-Aided Des. Comput. Graph.* **2015**, *27*, 9–25.
5. You, W.; Dong, C.; Wu, Q.; Qu, Y.; Wu, Y.; He, R. Joint task scheduling, resource allocation, and UAV trajectory under clustering for FANETs. *China Commun. Engl.* **2022**, *19*, 15. [[CrossRef](#)]
6. Su, J.; He, X.; Qing, L.; Niu, T.; Cheng, Y.; Peng, Y. A novel social distancing analysis in urban public space: A new online spatio-temporal trajectory approach. *Sustain. Cities Soc.* **2021**, *68*, 102765. [[CrossRef](#)]
7. Wang, W.; Xie, Y. Multi-Level Clustering Algorithm for Pedestrian Trajectory Flow Considering Multi-Camera Information. In Proceedings of the 2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), Virtual, 23–25 September 2022; pp. 691–698.
8. Charou, E.; Kabassi, K.; Martinis, A.; Stefanouli, M. Integrating multimedia GIS technologies in a recommendation system for geotourism. In *Multimedia Services in Intelligent Environments*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 63–74.
9. McDermid, G.J.; Franklin, S.E.; LeDrew, E.F. Remote sensing for large-area habitat mapping. *Prog. Phys. Geogr.* **2005**, *29*, 449–474. [[CrossRef](#)]
10. Navarrete, T.; Blat, J. VideoGIS: Segmenting and indexing video based on geographic information. In Proceedings of the 5th AGILE Conference on Geographic Information Science, Palma, Spain, 25–27 April 2002; p. 9.
11. Han, Z.; Kong, Y.; Qin, Q.; Wang, W. Geographic stereo video data analysis and model design. *Geogr. Geo-Inf. Sci.* **2013**, *29*, 1–7.
12. Feng, J.; Song, H. Analytical method for mobile elements in geo-video using random graph grammar. *Geomat. Inf. Sci. Wuhan Univ.* **2014**, *39*, 206–209.
13. Xie, Y.; Wang, M.; Liu, X.; Mao, B.; Wang, F. Integration of multi-camera video moving objects and GIS. *Int. J. Geo-Inf.* **2019**, *8*, 561. [[CrossRef](#)]
14. Milosavljević, A.; Rančić, D.; Dimitrijević, A.; Predić, B.; Mihajlović, V. A Method for Estimating Surveillance Video Georeferences. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 211. [[CrossRef](#)]
15. Lewis, P.; Fotheringham, S.; Winstanley, A. Spatial video and GIS. *Int. J. Geogr. Inf. Sci.* **2011**, *25*, 697–716. [[CrossRef](#)]
16. Walton, S.; Berger, K.; Ebert, D.; Chen, M. Vehicle object retargeting from dynamic traffic videos for real-time visualization. *Vis. Comput.* **2014**, *30*, 493–505. [[CrossRef](#)]
17. Du, R.; Bista, S.; Varshney, A. Video fields: Fusing multiple surveillance videos into a dynamic virtual environment. In Proceedings of the 21st International Conference on Web3D Technology, Anaheim, CA, USA, 22–24 July 2016; pp. 165–172.
18. Wu, C.; Zhu, Q.; Zhang, Y.; Du, Z.; Zhou, Y.; Xie, X. An adaptive organization method of geovideo data for spatio-temporal association analysis. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *29*. [[CrossRef](#)]

19. Cho, Y.; Park, J.; Kim, S.; Le, K.; Yoon, K. Unified framework for automated person re-identification and camera network topology inference in camera networks. *arXiv* **2017**, arXiv:1704.07085.
20. Jian, H.; Liao, J.; Fan, X.; Xue, Z. Augmented virtual environment: Fusion of real-time video and 3D models in the digital earth system. *Int. J. Digit. Earth* **2017**, *10*, 1177–1196. [CrossRef]
21. Loy, C.; Xiang, T.; Gong, S. Time-delayed correlation analysis for multi-camera activity understanding. *Int. J. Comput. Vis.* **2010**, *90*, 106–129. [CrossRef]
22. Mehboob, F.; Abbas, M.; Rehman, S.; Khan, S.A.; Jiang, R.; Bouridane, A. Glyph-based video visualization on Google Map for surveillance in smart cities. *EURASIP J. Image Video Process.* **2017**, *2017*, 28. [CrossRef]
23. Chen, Z.; Shen, H.T.; Zhou, X.; Zheng, Y.; Xie, X. Searching trajectories by locations: An efficiency study. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, Indianapolis, IL, USA, 6–10 June 2010; pp. 255–266.
24. Gurung, S.; Lin, D.; Jiang, W.; Hurson, A.; Zhang, R. Traffic information publication with privacy preservation. *ACM Trans. Intell. Syst. Technol. (TIST)* **2014**, *5*, 44. [CrossRef]
25. Yao, T.; Wang, Z.; Xie, Z.; Gao, J.; Feng, D.D. Learning universal multiview dictionary for human action recognition. *Pattern Recognit.* **2017**, *64*, 236–244. [CrossRef]
26. Zhao, W.; Zhang, Z.; Huang, K. Gestalt laws based tracklets analysis for human crowd understanding. *Pattern Recognit.* **2018**, *75*, 112–127. [CrossRef]
27. Kumar, S.; Dai, Y.; Li, H. Spatio-temporal union of subspaces for multibody non-rigid structure-from-motion. *Pattern Recognit.* **2017**, *71*, 428–443. [CrossRef]
28. Nanni, M.; Pedreschi, D. Time-focused clustering of trajectories of moving objects. *J. Intell. Inf. Syst.* **2006**, *27*, 267–289. [CrossRef]
29. Li, X.; Hu, W.; Hu, W. A coarse-to-fine strategy for vehicle motion trajectory clustering. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR’06), Hong Kong, China, 20–24 August 2006; Volume 1, pp. 591–594.
30. Ferreira, N.; Kłosowski, J.T.; Scheidegger, C.E.; Silva, C.T. Vector field k-means: Clustering trajectories by fitting multiple vector fields. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2013; Volume 32, pp. 201–210.
31. Yuan, Y.; Feng, Y.; Lu, X. Statistical hypothesis detector for abnormal event detection in crowded scenes. *IEEE Trans. Cybern.* **2017**, *47*, 3597–3608. [CrossRef]
32. Wang, L.; Dong, M. Detection of abnormal human behavior using a matrix approximation-based approach. In Proceedings of the 2014 13th International Conference on Machine Learning and Applications, Detroit, MI, USA, 3–5 December 2014; pp. 324–329.
33. Wang, R.; Zheng, W.; Huang, M.; Li, G. Driving Behavior Evaluation Based on DBSCAN and Kmeans++ Clustering. In Proceedings of the 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Wuhan, China, 22–24 April 2022; pp. 188–193.
34. Yao, D.; Hu, H.; Du, L.; Cong, G.; Han, S.; Bi, J. Trajgat: A graph-based long-term dependency modeling approach for trajectory similarity computation. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 2275–2285.
35. Park, J.; Jeong, J.; Park, Y. Ship trajectory prediction based on bi-LSTM using spectral-clustered AIS data. *J. Mar. Sci. Eng.* **2021**, *9*, 1037. [CrossRef]
36. Zheng, Y.; Zhang, L.; Xie, X.; Ma, W.Y. Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of the 18th International Conference on World Wide Web, Madrid, Spain, 20–24 April 2009; pp. 791–800.
37. Guha, S.; Rastogi, R.; Shim, K. CURE: An efficient clustering algorithm for large databases. *ACM Sigmod Rec.* **1998**, *27*, 73–84. [CrossRef]
38. Zhang, L.; Zhu, Y.; Su, J.; Lu, W.; Li, J.; Yao, Y. A Hybrid Prediction Model Based on KNN-LSTM for Vessel Trajectory. *Mathematics* **2022**, *10*, 4493. [CrossRef]
39. Wu, J.; Cai, S.; Jin, H.; Liu, L. Vehicular delay tolerant network routing algorithm based on trajectory clustering and dynamic Bayesian network. *Wirel. Netw.* **2023**, *29*, 1873–1889. [CrossRef]
40. Zeng, W.; Xu, Z.; Cai, Z.; Chu, X.; Lu, X. Aircraft trajectory clustering in terminal airspace based on deep autoencoder and gaussian mixture model. *Aerospace* **2021**, *8*, 266. [CrossRef]
41. Zhong, G.; Zhang, H.; Zhou, J.; Zhou, J.; Liu, H. Short-Term 4D Trajectory Prediction for UAV Based on Spatio-Temporal Trajectory Clustering. *IEEE Access* **2022**, *10*, 93362–93380. [CrossRef]
42. Aparna, R.; Idicula, S.M. Spatio-temporal data clustering using deep learning: A review. In Proceedings of the 2022 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS), Larnaca, Cyprus, 25–26 May 2022; pp. 1–10.
43. Li, Q.; He, X.; Chen, K.; Ouyang, Q. A Two-Stage Semi-Supervised High Maneuvering Target Trajectory Data Classification Algorithm. *Appl. Sci.* **2022**, *12*, 10979. [CrossRef]
44. Ferreira, M.D.; Spadon, G.; Soares, A.; Matwin, S. A semi-supervised methodology for fishing activity detection using the geometry behind the trajectory of multiple vessels. *Sensors* **2022**, *22*, 6063. [CrossRef]
45. Ristani, E.; Solera, F.; Zou, R.; Cucchiara, R.; Tomasi, C. Performance measures and a data set for multi-target, Multi-Camera Tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
46. Kim, K.; Oh, S.; Lee, J.; Essa, I. Augmenting aerial earth maps with dynamic information from videos. *Virtual Real.* **2011**, *15*, 185–200. [CrossRef]
47. Kumawat, M.; Khaparde, A. Development of adaptive time-weighted dynamic time warping for time series vegetation classification using satellite images in Solapur district. *Comput. J.* **2022**, bxac057. [CrossRef]

48. Cao, Y.; Tang, K.; Sun, J.; Ji, Y. Day-to-day dynamic origin–destination flow estimation using connected vehicle trajectories and automatic vehicle identification data. *Transp. Res. Part C* **2021**, *129*, 103241. [[CrossRef](#)]
49. Xi, J.; Jia, F.; Feng, J. An online estimation method for passenger flow OD of urban rail transit network by using AFC data. *J. Transp. Syst. Eng. Technol.* **2019**, *18*, 129–135.
50. Tang, Z.; Naphade, M.; Liu, M.Y.; Yang, X.; Birchfield, S.; Wang, S.; Kumar, R.; Anastasiu, D.; Hwang, J.N. Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8797–8806.
51. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
52. Zhao, H.; Gao, J.; Lan, T.; Sun, C.; Sapp, B.; Varadarajan, B.; Shen, Y.; Shen, Y.; Chai, Y.; Anguelov, D. Tnt: Target-driven trajectory prediction. In Proceedings of the Conference on Robot Learning, London, UK, 8–11 November 2021; pp. 895–904.
53. Hou, X.; Wang, Y.; Chau, L.P. Vehicle tracking using deep sort with low confidence track filtering. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, China, 18–21 September 2019; pp. 1–6.
54. Zheng, Z.; Zheng, L.; Yang, Y. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3754–3762.
55. Tekler, Z.D.; Low, R.; Gunay, B.; Andersen, R.K.; Blessing, L. A scalable Bluetooth Low Energy approach to identify occupancy patterns and profiles in office spaces. *Build. Environ.* **2020**, *171*, 106681. [[CrossRef](#)]
56. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [[CrossRef](#)]
57. Yu, X.; Long, W.; Li, Y.; Gao, L.; Shi, X. Trajectory dimensionality reduction and hyperparameter settings of DBSCAN for trajectory clustering. *IET Intell. Transp. Syst.* **2022**, *16*, 691–710. [[CrossRef](#)]
58. Salvador, S.; Chan, P. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.* **2007**, *11*, 561–580. [[CrossRef](#)]

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.