International Journal of Geo-Information

MDPI

Article

A Two-Step Clustering Approach to Extract Locations from Individual GPS Trajectory Data

Zhongliang Fu [1,2], Zongshun Tian [1,*], Yanqing Xu [3] and Changjian Qiao [1]

[1] School of Remote Sensing and Information Engineering, Wuhan University, No. 129 Luoyu Road, Wuhan 430079, China; fuzl@whu.edu.cn (Z.F.); 2011202130013@whu.edu.cn (C.Q.)
[2] Collaborative Innovation Center of Geospatial Technology, No. 129 Luoyu Road, Wuhan 430079, China
[3] Department of Geography & Planning, University of Toledo, 2801 W. Bancroft, Toledo, OH 43606, USA; xuyq@utoledo.edu
* Correspondence: tzshun@whu.edu.cn; Tel.: +86-137-2024-1115

Academic Editor: Wolfgang Kainz
Received: 26 July 2016; Accepted: 19 September 2016; Published: 23 September 2016

Abstract: High-accuracy location identification is the basis of location awareness and location services. However, because of the influence of GPS signal loss, data drift and repeated access in the individual trajectory data, the efficiency and accuracy of existing algorithms have some deficiencies. Therefore, we propose a two-step clustering approach to extract individuals' locations according to their GPS trajectory data. Firstly, we defined three different types of stop points; secondly, we extracted these points from the trajectory data by using the spatio-temporal clustering algorithm based on time and distance. The experimental results show that the spatio-temporal clustering algorithm outperformed traditional extraction algorithms. It can avoid the problems caused by repeated access and can substantially reduce the effects of GPS signal loss and data drift. Finally, an improved clustering algorithm based on a fast search and identification of density peaks was applied to discover the trajectory locations. Compared to the existing algorithms, our method shows better performance and accuracy.

Keywords: GPS trajectory data; data mining; clustering algorithm; personal location

1. Introduction

More and more personal daily trajectory data are being recorded with the popularity of smart mobile terminals. How to extract useful information from these trajectory data quickly and accurately in order to provide personalized location services is the primary concern of location-aware computing. Through the analysis of a large amount of data, Gonzalez et al. [1] found that human trajectories show a high degree of temporal and spatial regularity. A user has high frequency access to certain known locations, which means location is key information for trajectory data. Personal location is the critical component of location-based services [2]. By accurately identifying the individual location from the trajectory data, one can make location-aware applications more intelligent [3]. For example, the navigator can guide the user with personal locations [4]. Also, the location-based recommender system can push information relevant to the location of people [5]. A location prediction system can also speculate an individual's next destination according to their current position [6], and a spatial query system can provide contextualized information [7], etc.

There is a rich body of research on how to discover personal location from trajectory data. For the wireless beacon trajectory data, such as data based on wireless networks or cell towers, some characteristics of the beacon (e.g., the access point information or signal response rate) may be recorded as a fingerprint to identify the location in the future [8,9]. Compared to GSM or Wi-Fi positioning, GPS is more accurate [10]. Therefore, we focus on the processing of GPS trajectory data in this paper.

One characteristic of the individual trajectory data is that it repeatedly accesses the same reference point. Meanwhile, affected by the existing GPS technology, trajectory data suffers from two drawbacks: first, the discontinuity caused by GPS signal loss; second, the data drift caused by other environmental factors. These problems have a significant influence on the location extraction algorithms. The existing algorithms are mostly based on temporal and spatial characteristics of the trajectory data, or they are in combination with the thematic data in order to find the location. These algorithms have their advantages, but there are some deficiencies when dealing with the above mentioned problems.

The most direct way is clustering GPS points based on the point density. For example, Schuessler & Axhausen [11] used the density threshold to identity the locations. Ashbrook & Starner [12] used a variation of the K-Means clustering algorithm to learn locations from trajectory data. Zhou et al. [13] proposed the DJ-Cluster algorithm based on the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [14] in order to recognize the location. These algorithms do not provide a processing capacity for the points with signal loss and data drift, especially the signal loss caused by a building block. They only consider the spatial position regardless of the time attribute. Therefore, these algorithms are only suitable for the points with strong signal.

To solve the problems caused by data loss, Ashbrook & Starner [15] proposed to firstly identify the lost points. Then, they used the improved K-Means algorithm to cluster the lost points in order to form the locations. Since the K-Means algorithm is sensitive to noise, and the approach only recognizes the lost points, performance is poor. Therefore, more algorithms select the location based on time and space characteristics. For example, Du & Aultman-Hall [16] considered the sequential GPS points with a lower speed as a location. Kang et al. [17] proposed a time-based clustering algorithm. The algorithm takes these points, which are close to each other, and the time interval, which is greater than the threshold at the locations. The SMoT algorithm [18] did an analysis based on the intersection of trajectory data and candidate stops, in which overlapped and sustained points were considered as locations. Palma et al. [19] proposed the CB-SMoT algorithm based on the previous studies. The algorithm clustered data with the parameters Eps (the radium of the cluster) and the minimum time based on the DBSCAN algorithm. Then, it analyzed the intersection of those clusters and the candidate stops. To some extent, these algorithms can avoid the problems caused by data loss. However, due to the impact of data drift, these algorithms are prone to false positives. These algorithms also do not consider the problem of repeated access to the same place. Consequently, each location which was extracted by these algorithms will be considered as a new location.

Therefore, a hierarchical clustering algorithm was applied to location extraction. Yuan et al. [20] proposed an approach which firstly divided the trajectory data into segments, and then clustered the endpoints of the segments to the locations. Although the algorithm overcomes the problem of repeated access, the result is not ideal due to the impact of signal loss and data drift. More importantly, the endpoints of the segments only represent the main turning point of the trajectory. To mitigate these drawbacks, Lv et al. [3] used a time-based algorithm to obtain the visit points which were then clustered to form physical places by an improved DBSCAN algorithm. Experiments have achieved good results, but the algorithm also has some problems. First, influenced by the data drift, a visit point is easily divided into multiple small clusters. Consequently, the precision and recall rate of the algorithm are affected by the segmentation. Second, the DBSCAN algorithm is very sensitive to the parameters [21] and has a relatively high time complexity. It is not suitable for processing a large amount of data as part of the trajectory data mining.

With such concerns about the previously used algorithms, we propose a two-step clustering approach to extract personal locations from individual GPS trajectory data. First, we extracted the possible stop points by using the spatial-temporal clustering algorithm. Considering the characteristics of the individual trajectory data, this paper presents three different types of stop points, among which the first type is not taken into account in the existing algorithm. The spatio-temporal clustering algorithm can largely avoid the problems of signal loss and data drift. It can also reduce the impact of segmentation found in the literature [3]. Next, an improved fast clustering algorithm is adopted

to cluster the stop points into the locations; it can greatly improve efficiency and accuracy. Finally, we designed the contrasting experiments to verify the feasibility and effectiveness of the approach.

2. Location Extraction

As shown in Figure 1, a trajectory is composed of GPS points in chronological order, and different trajectories in daily life constitute the dataset. Considering the characteristics of an individual trajectory, our approach adopts a spatio-temporal clustering based on time and distance to identify stop points from the dataset (more details in Section 2.1), then extracts locations from these stop points based on an improved clustering by a fast search and identification of density peaks (more details in Section 2.2).

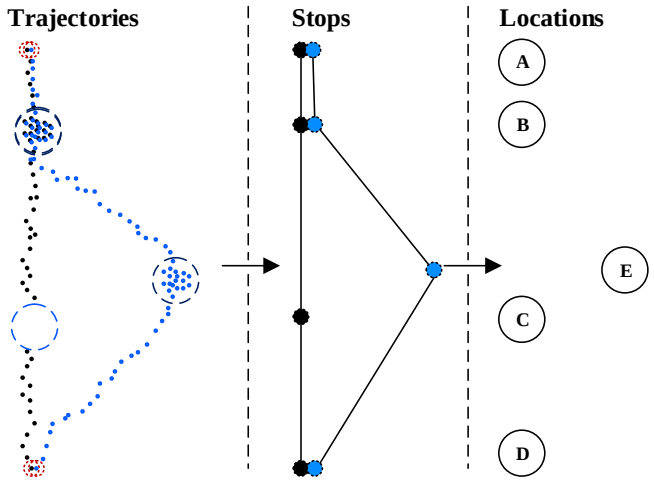

Figure 1. The flow chart of location extraction.

Definition 1. ***GPS point****: a GPS point is an attribute group. P = (lng, lat, alt, d, t, v, satellites, HDOP) represents the longitude, latitude, altitude, date, time, speed, and number of tracked satellites and the position dilution of precision.*

Definition 2. ***Trajectory****: a trajectory is a list of GPS points based on their time serials, $T = \{P_0, P_1, \ldots, P_n\}$, where $t_0 < t_1 < \ldots < t_n$ and $i = 0, 1, \ldots, n$. Usually, a trajectory consists of GPS points which are collected in one day.*

Definition 3. ***Stop Point****: a stop point stands for a visited location where a person arrived at t_{start} and left at t_{end}, $SP = \{P_m, P_{m+1}, \ldots, P_n\} = (lng, lat, type, t_{start}, t_{end})$, (lng, lat) represents the mean coordinate of points and type represents the type of a stop point. It depends on two parameters, the time threshold δ_t and the distance threshold δ_d (more details in Section 2.1).*

Definition 4. ***Location Point****: a location point means a location that a person frequently visits, $LP = \{SP_i, \ldots, SP_j, \ldots, SP_k\}$. It comes from the results of a stop point clustering algorithm (more details in Section 2.2).*

2.1. Extracting Stop Points

As mentioned above, because of the influence of GPS signal loss and data drift, most of the present algorithm of extracting stop points has some shortcomings. To counter this problem, we propose a novel spatio-temporal clustering based on time and distance, which is abbreviated to TDBC. The algorithm has a better adaptation for individual trajectory when it extracts stop points.

In this paper, we define three different types of stop points.

The first type comes from the start or end point of each track. Trajectory data acquisition starts from a location where a person leaves (e.g., leave home to work in the morning) and ends in a position

for a long time without a signal (e.g., arrive home after work in the evening). However, taking into account the occasional occurrence of data acquisition, a trajectory may end up with other factors, such as a low battery or man-made factor. Therefore, not all start or end points are the first type of stop point. Only those points which are adjacent to each other will be considered the first type, (type I, e.g., Point a in Figure 2). A type I stop point has two points in $T = (P_0, P_1, \ldots, P_n)$, where distance $(P_0, P_n) < \delta_d$.

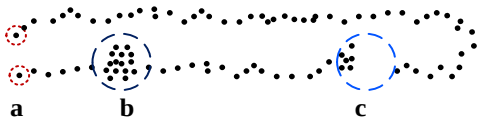

Figure 2. The different types of stop points. (**a**) Type I; (**b**) Type II; (**c**) Type III.

When a person stays in one place for a long time without a signal loss, the trajectory points form the second type (type II, e.g., Point b in Figure 2). Our algorithm depends on two parameters, the time threshold δ_t and the distance threshold δ_d. A type II stop point is a list of consecutive GPS points, $SP = \{P_m, P_{m+1}, \ldots, P_n\}$, where $\forall m \leq i \leq n$, distance$((lng, lat), P_i) < \delta_d$ and $|P_n.t - P_m.t| > \delta_t$.

The third type is the region where signal is lost for a long time due to barriers, buildings or equipment problems, etc. (type III, e.g., Point c in Figure 2). It needs the two parameters as well. A type III stop point has two sequential GPS points $SP = \{P_m, P_{m+1}\}$, where distance $(P_m, P_{m+1}) < \delta_d$ and $|P_{m+1}.t - P_m.t| > \delta_t$.

As shown in Algorithm 1, we adopt a spatio-temporal clustering method, which is named TDBC, to extract stop points from the individual trajectory. The TDBC takes single trajectory T, time threshold δ_t and distance threshold δ_d as input conditions. *Cluster* and *Previous C* are the current cluster and the previous cluster which are used to deal with the second type of stop point.

The TDBC processes the GPS points along the time axis for each individual trajectory. According to the definition of the type, the TDBC takes these points which conform to the definition requirements as stop points, and puts them into the set SP (type I, line 2; type II, line 6; type III, line 10).

If the distance between the centroid of *Cluster* and the point P_i is greater than δ_d, and the time duration is less than δ_t, the TDBC does not ignore the *Cluster*, but checks the relationship between *Cluster* and *Previous C* (as shown in ***Function*** check). This strategy can effectively avoid the problem, such as when a stop point is divided into several clusters, which may stem from the precision of the GPS device or data drift.

The function *SP*.add (*Cluster*) in the algorithm means that the function considers the *Cluster* as a stop point and puts in the set *SP*. However, the operation is not done until the function checks whether or not the *Cluster* can merge with the *Previous stop point*. The two clusters are merged if the condition is met, otherwise the function puts a new stop point in the set *SP*. The purpose of this function is to merge the stop points which have spatial neighbors and continuous time relationships with each other. A stop point, which stays within a certain range for an extended period, is not divided into several parts by this method. Therefore, it can improve the accuracy of extracting stop points.

As is apparent from the above description, the TDBC examines all the GPS points only once. Therefore, the time complexity of the TDBC is $O(n)$. It has an excellent ability to handle the consecutive GPS points with data drift (e.g., stay in a partly closed building) and avoids the inefficient problems caused by the segmentation of a stop point.

Algorithm 1. Stop Point Extraction (TDBC)

Input: Trajectory T, time threshold δ_t, distance threshold δ_d

Output: a set of stop points SP

```
1:  Cluster=∅; Previous C=∅;
2:  if the first or last point is type I then SP.add(the point);   // type I
3:  for each point P_i in T do
4:    if distance(Cluster, P_i) < δ_d then put P_i in Cluster; continue;
5:    if distance(Cluster, P_i) > δ_d and duration(Cluster) > δ_t then
6:      SP.add(Cluster); continue;                                 // type II
7:    if distance(Cluster, P_i) > δ_d and duration(Cluster) < δ_t then
8:      check(Cluster, Previous C); continue;
9:    if distance(P_i-1, P_i) < δ_d and duration(P_i-1, P_i) > δ_t then
10:     SP.add({P_i-1, P_i}); continue;                            // type III
11:   if distance(P_i-1, P_i) > δ_d and duration(P_i-1, P_i) > δ_t then ignore();
12: return SP;

    Function check(Cluster, Previous Cluster){
      if (time interval(Cluster, Previous) < δ_t and distance(Cluster, Previous) < δ_d) then
        Previous = merge(Cluster, Previous);
        if Previous is one of type II then SP.add(Previous);
          else Previous = Cluster;}

    Function SP.add(Cluster){
      if (distance(Cluster, Previous stop point in SP) < δ_d) then
      Previous = merge(Cluster, Previous);
        else put Cluster in SP; Previous = Cluster;}
```

2.2. Extracting Locations

A location point means a person frequently visits the location. The personal location information is a critical component in the field of location awareness, location prediction, social recommendations, etc. It varies significantly in order to identify the individual's location. As mentioned in Section 2.1, the algorithm of TDBC is designed to extract stop points from a single trajectory. However, a person repeatedly visits the same location at a different time, and due to the possible lack of precision of the GPS equipment, it is necessary to cluster the stop points to obtain the location points.

2.2.1. Clustering by Fast Search and Find of Density Peak

Currently, many clustering algorithms are applicable to the problem. The literature has proposed an approach based on clustering by a fast search and identification of density peaks [22]. The algorithm is very efficient. It assumes that cluster centers are at a relatively large distance from any points which have a higher density, as well as centers that are surrounded by points which have a lower density. The assumption fits well with the data distribution characteristics of a stop point. Generally speaking, different stop points in the same location are not far away from each other, and different locations are not close to each other.

Definition 5. ***Density****: the density ρ_i of point SP_i is the number of the points, which the distance between the point SP_j and SP_i is less than δ_d, defined as Equation (1).*

$$\rho_i = \sum_j f\left(d_{ij} - \delta_d\right) \tag{1}$$

where if $x \leq 0$, $f(x) = 1$, otherwise $f(x) = 0$. d_{ij} means the distance between SP_j and SP_i. When $i = j$, $d_{ij} = 0$.

Definition 6. ***Neighbor****: the neighbor N_i = {SP$_j$, . . . , SP_k, . . . , SP_l} of point SP_i represents the set of stop points, whereby the distance between the current stop point SP_i and the others is less than δ_d.*

Algorithm flow is as follows:

Step 1, calculate the density and the neighbor for each point, and the points must be listed in descending order by the density.

Step 2, calculate the minimum distance between the point and other points with higher local density, Equation (2).

$$\delta_i = \min_{j:\rho_j > \rho_i} \left(d_{ij}\right). \tag{2}$$

If the density is the maximum, the distance is defined as $\delta_i = \max(d_{ij})$. Thus, for the cluster centers, the value of δ_i is obviously larger than that of the neighbors.

Step 3, calculate $\lambda_i = \rho_i \delta_i$ for each point and sort the results in decreasing order. These points, which have relatively larger λ_i, are the cluster centers.

Step 4, according to the density value of descending order, assign the non-central points to its nearest neighbor of higher density.

Compared with the optimized iteratively algorithms, it can accurately and quickly find clusters of arbitrary shapes. After the cluster centers are found, the cluster assignment is performed in a single step. It can be well applied to extract locations from stop points.

2.2.2. Problem Interpretation and Algorithm Improvement

The clustering, by conducting a fast search and identification of density peaks, is suitable for irregularly scattered points clustering. The quantity of points, which has the same density, is not significant. The algorithm is not significantly affected when the cluster centers are determined. However, this is not the case for trajectory data, different stop points in the same location are not far away from each other, having many stop points with the same density. If we use the algorithm without any improvements, a large number of error cluster centers will be selected. As shown in Figure 3, there are three places, A, B, and C, and a person visited them 5 times, 4 times, and 4 times, respectively. According to Step 2, the distance value of the four stop points in B is approximately equal to d, in Figure 3b. Finally, the four stop points are selected as the centers, which are obviously wrong, in Figure 3c. The red triangle represents the cluster center.

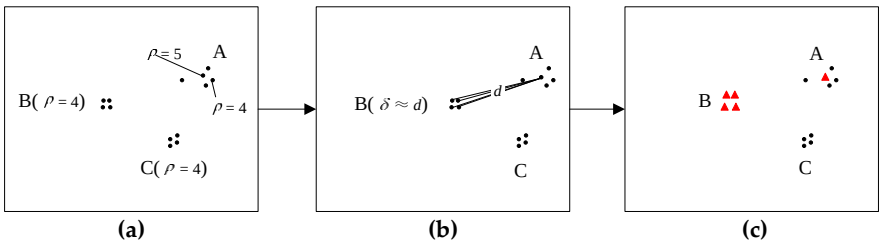

Figure 3. An example of wrong clustering centers. (**a**) The initial points; (**b**) For each point, calculate the minimum distance; (**c**) Choose the wrong centers.

In order to solve the problem concerning the same density as mentioned above, the density can be adjusted by the following steps. The algorithm takes the stop points which are sorted in descending order by the density as the input condition. The $\boldsymbol{a}$ represents the current density.

(1) $\boldsymbol{a} = \max(\rho)$.

(2) Take out the points with the same density $\boldsymbol{a}$, and process each point by the following steps 3 and 4.

(3) For point SP_i with the density $\boldsymbol{a}$, check the density of each point in neighbor N_i. If there is a point in N_i which the density is higher than $\boldsymbol{a}$, ignore the point SP_i, otherwise put it into the set $S(\rho = \boldsymbol{a})$.
(4) For each point with the density $\boldsymbol{a}$ in neighbor N_i, repeat Steps 3 and 4 until all the points with the density $\boldsymbol{a}$ have been processed and form the set $S(\rho = \boldsymbol{a}) = \{SP_i, \ldots, SP_j, \ldots, SP_k\}$.
(5) For each point in the set S, check the neighbor relationship with each other. If two points are neighborhood, put them into a subset SB_j. Then, form the set $S = \{SB_1, \ldots, SB_m\}$, $SB_j = \{SP_l, \ldots, SP_k\}$, which contains m subsets. For each subset, the points are adjacent to each other and have the same density $\boldsymbol{a}$.
(6) For each subset SB_i, pick a point randomly and make its density equal to $\rho_i = \boldsymbol{a} + i/(m+1)$.
(7) $\boldsymbol{a} = \boldsymbol{a} - 1$, repeat Steps 2 to 7. If $\boldsymbol{a} = 1$, the algorithm is terminated.

Figure 4 demonstrates the whole process of adjusting the density. In Figure 4a, there is a point with a density of 4 in A. However, in the neighbor of the point, there is a point with a density of 5. Therefore, we ignore the point. In Figure 4b, the points in B and C form two subsets, and two points are given a new density value, according to the above process. In Figure 4c, the cluster centers are selected by the new density.

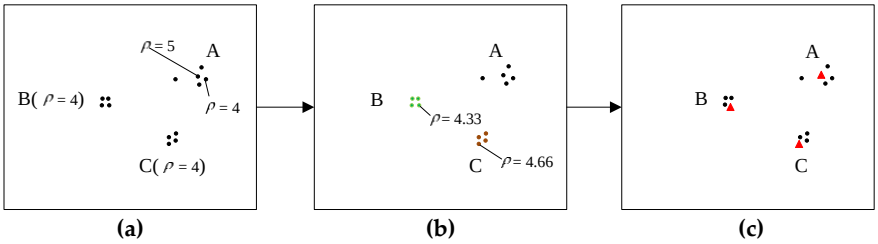

Figure 4. An example of adjusting the density. (**a**) The initial points; (**b**) For the points which have the same density, recalculate the density; (**c**) Choose the centers.

Consider the particularity of the stop points: stop points with the density value that is equal to 1 indicates that the person has a single access record. That stop point is a cluster. Therefore, when the algorithm of clustering by a fast search and identification of density peaks calculates the minimum distance in Step 2, if the density is the maximum or 1, the distance should be recorded as $\delta_i = \max(d_{ij})$.

3. Experimental Section

3.1. Data Acquisition

In order to verify the effectiveness of our proposed algorithm, we invited 30 volunteers to collect their GPS trajectories in their daily lives. The experiment lasted for a month. All volunteers involved in the experiment are students and teachers from Wuhan University. The data collection application was installed in the volunteers' smart phones, which included Android or iOS. To ensure the accuracy of the data, the HOLUX GPSlim236 was taken as the GPS signal receiving the device and connected to the smart phone via Bluetooth. Therefore, GPS is the only GNSS in our experiment. As the HOLUX GPSlim236 updates data once per second, to reduce the redundant data, the application can adjust the sampling interval based on the mobile speed. The shortest interval is not less than 5 seconds.

During the trajectory data acquisition, we did not set up any restrictions or any manual intervention for volunteers. The volunteers were asked to carry out the GPS device in daylight to keep the application running in the background as much as possible, and the device was charged at night. More data was collected by the application to reflect the volunteers' daily lives. The trajectory data was stored in the memory of a smart phone on a daily timescale. The application not only can automatically record the track information, but also can manually mark the location. The volunteers were not forced to record location. When the volunteer moved into a place or during the period, the

volunteer should record the current coordinates and time through the application, which means the volunteer created a record of location. Each recorded location has a type attribute. In the application, there were five types for volunteers to choose from: home or dormitory, office or classroom, restaurant, entertainment (e.g., sight spot, cinema, etc.) and public service (e.g., hospital, bus station, etc.). In this paper, we defined the recorded location as $RL = (lng, lat, type, t)$, which represents a registered location's longitude, latitude, type of location and time. At the same time, if the volunteers record locations in the same place multiple times, the application will allow volunteers to choose the recorded locations and group them together. The recorded locations are mainly used for the verification of the stop point extraction algorithm, and the groups are for the location clustering algorithm.

During the whole experiment, we collected the trajectory data of 23 volunteers. Eleven of them recorded many locations. The acquired data shown in Table 1, Dataset 1 contains 11 volunteers' data over 20 days. The trajectory data is relatively complete. Datasets 2 and 3 are selected from Dataset 1.

Table 1. Details of different datasets.

	Raw Data	Filtered Data	Dataset 1	Dataset 2	Dataset 3
Volunteers	23	23	11	4	1
Days	26	22	20	1	1
GPS points	441,715	383,660	205,612	3726	1043
Recorded Locations	2863	2657	1541	31	9

3.2. Preprocessing

Because of the influence of GPS signal loss and data drift, there are a number of outliers in the trajectory data during the data acquisition. Therefore, the data needs to be cleaned. According to these parameters which reflect the GPS signal quality such as the altitude, number of tracked satellites, HDOP and speed, the raw data can be filtered. Wuhan is located in Jianghan Plain, averaging below 50 m above sea level. The region of data acquisition mainly focused in Wuhan University. Considering the GPS vertical positioning accuracy, the altitude threshold is set to 100 m. These outliers, which are caused by the altitude data abnormality, can be filtered through experiments. This article draws on Stopher's research results [5]. In the algorithm of preprocessing, the number of tracked satellites is set to 3 and the HDOP is set to 5 to remove the points of large data drift. Taking into account the different travel habits of volunteers, the speed threshold is not configured as a fixed value. For each volunteer's data, a random sample of the trajectory data is used to determine the speed threshold by the Pauta criterion $\delta_v = \mu + 3\sigma$.

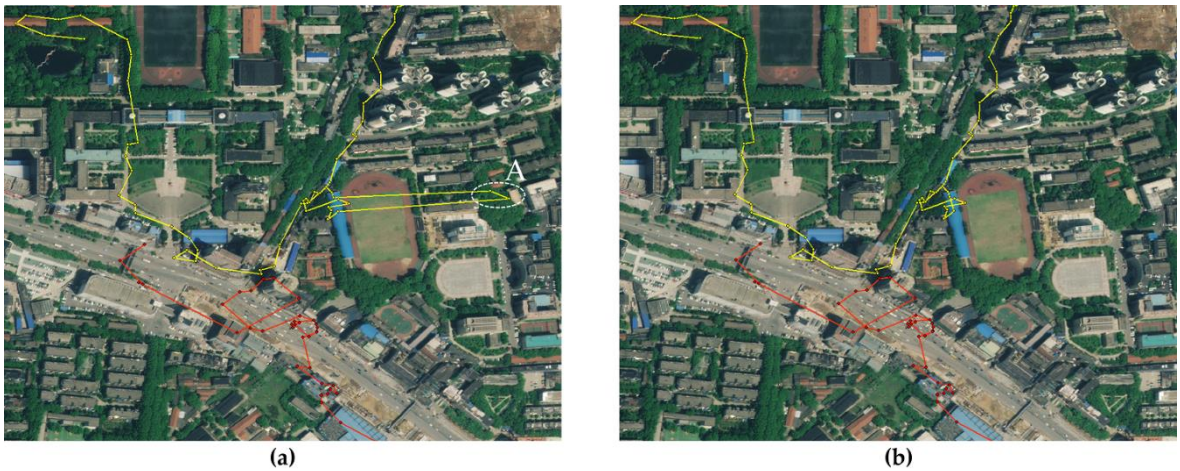

Figure 5. (**a**) Raw data; (**b**) Filtered data.

The purpose of Algorithm 2 is to filter outliers from the raw data. For each GPS point in the trajectory data, the point's speed is equal to the distance between the point and the next point divided by the time duration. This is because there are some sequential outliers in the raw data while the filtering parameters are standard, just as the area A in Figure 5a. The yellow and red lines represent the different GPS trajectories, which were collected in various routes. In total, 13.14% of the raw data was filtered after the preprocessing. Even though there are still outliers in the remaining data, we can reduce the influence of these outliers by using the point buffer.

Algorithm 2. Pre-processing

Input: Trajectory T, speed threshold δ_v, altitude threshold δ_{alt}, satellites threshold $\delta_{satellites}$, HDOP threshold δ_{hdop}

Output: T without outliers

previous point = P_0;
for each point P_i **in** T **do**
  v_i = distance(*previous point*, P_i) / duration(*previous point*, P_i);
  if $v_i > \delta_v$ **or** $P_i.alt > \delta_{alt}$ **or** $P_i.satellites < \delta_{satellites}$ **or** $P_i.HDOP > \delta_{hdop}$ **then**
    remove P_i from T;
  else *previous point* = P_i;
return T;

3.3. Parameters

Before the verification of the algorithm, we need to select the parameters of the algorithm. As we mentioned above, there are two parameters: the time threshold and the distance threshold.

In the algorithm of TDBC, the time threshold parameter δ_t determines the number of the clusters, because the algorithm considers a cluster as a stop point only when the time duration of the cluster is over δ_t. If the δ_t is too large, these clusters, which have a relatively short time duration, are ignored. Otherwise, these clusters are considered as the stop points and are joined in the set.

The distance threshold parameter δ_d determines the size of the clusters. For each point, the algorithm calculates the distance between the point and the centroid of the current cluster. The point joins the current cluster only when the distance is less than δ_d. A greater setting of δ_d results in a merge of clusters, which are adjacent to each other. The lesser setting of δ_d gives smaller clusters, which may be regarded as noise. Due to the lack of precision of GPS equipment, even if a person stays in the same place for a long time, the GPS data cannot be the same. If the δ_t is less than the level of precision, the algorithm produces some segmented clusters from an extended period during a visit. These clusters may be missed because the time duration is not sufficient to cover the δ_t.

Meanwhile, the distance threshold has effects on the location clustering algorithm. Individual location is a subjective concept which has different definitions according to different scales. Considering the application of location (e.g., navigation and location-based recommender), the location should be based on the building level. Therefore, the distance threshold value should not be too large.

In order to find the appropriate range of parameters, this paper selects Dataset 2 as the experimental data, which contains four volunteers' trajectories in one day. The graphs in Figure 6 show the number of stop points extracted from Dataset 2 for different distance and time thresholds.

From the graphs, we can observe that the curves decrease intensely when δ_t increases from 0 s to 300 s. The short time threshold δ_t, results in a number of clusters with short durations that are treated as the stop points. When δ_t > 300 s, the curves become flat, which indicates that the influence of the time threshold is obviously weakened. Meanwhile, for different distance thresholds, the greater the time threshold, the more similar the number of stop points. Therefore, the time threshold δ_t should be selected from [300, 800] s.

Compared with the other distance thresholds, the curves decrease more rapidly with δ_d = 20 m. The reason for this is that the δ_d is close to the GPS equipment precision (the precision of the HOLUX GPSlim236 is 5–25 m). When δ_d = 200 m, the curves are too flat to recognize the stop points, especially for the building level of locations. When δ_d = 125 m, the curves are similar to 200 m, which means the distance threshold is large. For these distance thresholds, such as 50 m and 100 m, the curves have an obvious knee point. Therefore, the distance threshold δ_d should be selected from (20,125) m.

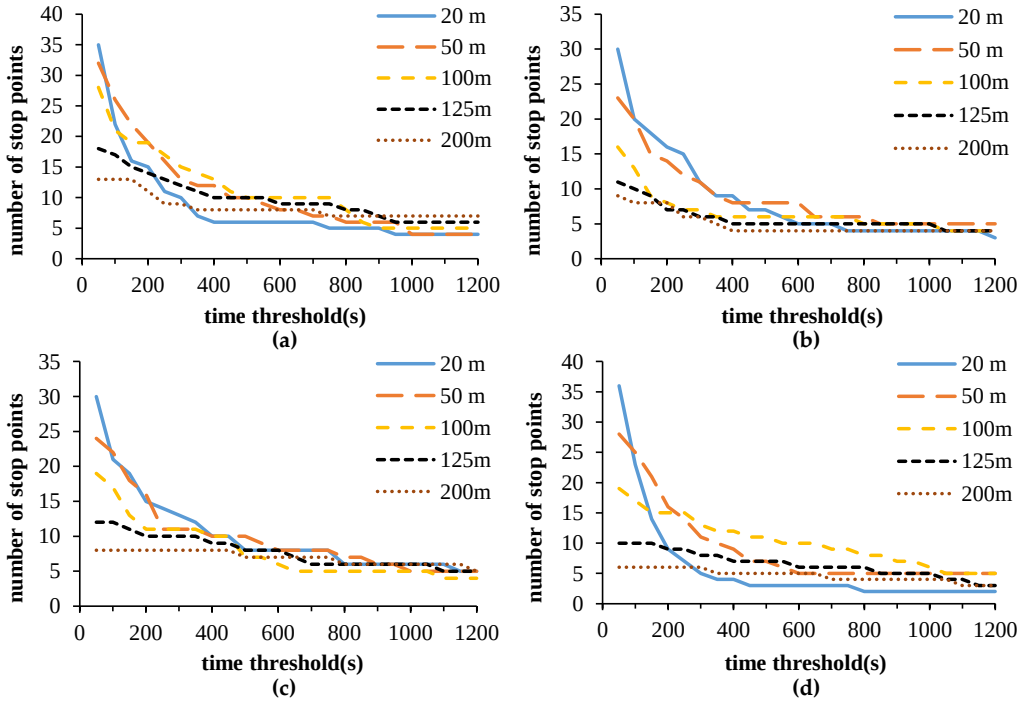

Figure 6. The number of stop points found for different distance and time thresholds. (**a**) The results of volunteer A; (**b**) Volunteer B; (**c**) Volunteer C; (**d**) Volunteer D.

To further estimate the parameters of the TDBC, this article uses the Precision (*P*), Recall (*R*), and F1-Measure (*F*) as the judging criteria. The recorded locations, which were recorded by volunteers, are regarded as a reference sample.

Definition 7. ***Match:*** *$\forall$a stop point $SP_i = (lng_i, lat_i, type_i, t_{i\,start}, t_{i\,end})$, $\exists$a recorded location $RL_j = (lng_j, lat_j, type_j, t_j)$, where $t_{i\,start} < t_j < t_{i\,end}$ and distance $(SP_i, RL_j) < \delta_d$. As shown in Figure 7, a matched point means the recorded point RL_1 is recorded during the stop point SP_1 and the distance between SP_1 and RL_1 is less than the distance threshold.*

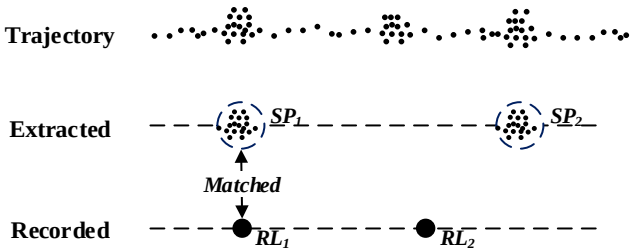

Figure 7. An example of the matched point.

The precision rate is the proportion of stop points, which match with the recorded locations, Equation (3). While p^+ stands for the number of the matched stop points, p^- does not. The recall rate is the proportion of recorded locations which are matched with the stop points, Equation (4). r^+ stands for the number of the matched recorded locations, while r^- does not.

$$P = \frac{p^+}{p^+ + p^-} \tag{3}$$

$$R = \frac{r^+}{r^+ + r^-} \tag{4}$$

We used Dataset 1 as the experiment data, which contains 11 volunteers' trajectory data and their recorded locations. They marked seven recorded locations per day on average. Figure 8 is the graphs of the precision and recall for different distance and time thresholds. When the precision rises, the recall decreases. On the whole, the reasonable precision rate is about 0.8 and the recall rate is about 0.7. A possible reason for this is that the recorded locations are not precise. For example, there were three visits by A, B, and C during a day and the duration times were 10, 15, and 30 mins. However, when the volunteer logged the location, he recorded one visit in B and two in C. Besides, recording locations was not mandatory during the data acquisition. Therefore, the Matched accuracy will be affected.

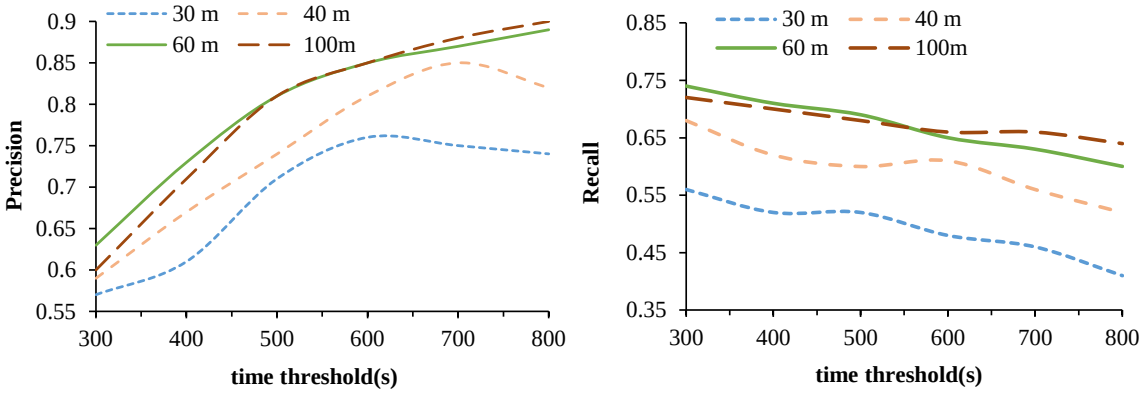

Figure 8. The precision and recall for different distance and time thresholds.

When the distance threshold δ_d is the same, the precision rate increases with the increase of the time threshold δ_t. This means that the stop points with a long duration, which are recognized by the TDBC, were just recorded by the volunteers. That is consistent with the actual situation. The longer a volunteer stays, the more likely to log a recorded location. Meanwhile, the recall rate decreases, which means the number of stop points reduces, and many stop points are not recognized.

When the time threshold δ_t is the same, both the precision and recall rates increase with the increase of the distance threshold δ_d. However, when δ_d exceeds 60 m, the precision and recall rates are not significantly improved, which indicates that a too small or too great of a distance parameter is not appropriate and δ_d = 60 m is the inflection point. To choose the appropriate parameters, the F1-Measure is further investigated.

F1-Measure is a comprehensive evaluation index based on the precision and recall rates.

$$F = \frac{2 \times P \times R}{P + R} \tag{5}$$

Figure 9 is the graph of the F1-Measurefor different distance and time thresholds. When δ_d is over 60 m, the F1-Measure remains constant. When δ_d = 60 m and δ_t = 500 s, the F1-Measure increases to the maximum. The precision rate is 0.81 and the recall rate is 0.68.

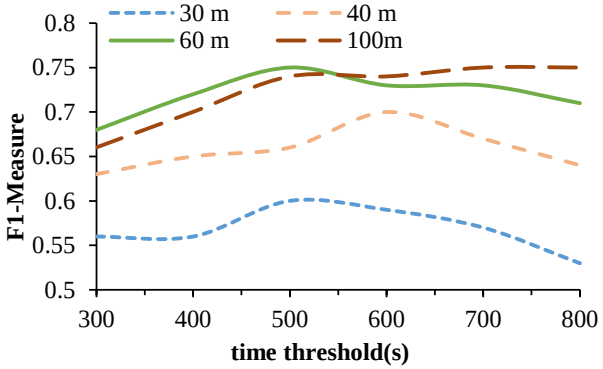

Figure 9. The F1-Measure for different distance and time thresholds.

3.4. Stop Point Extraction Experiment

In the following, two groups of experiments were designed to compare the TDBC with the other four algorithms, K-Medoids, DJ-Cluster [13], CB-SMoT [19] and Time-Based Clustering [3]. The first experiment used Dataset 1, which has a large amount of data, to compare the performance of the algorithms. The second experiment used Dataset 3 to compare the pros and cons of different algorithms under the specific conditions.

The results of the first experiment are listed in Table 2. The parameters of the TDBC are determined by the discussion in Section 3.3, while the other four algorithms selected parameters according to the optimal value of the Precision, Recall, and F1-Measure. For the CB-SMoT, we only completed the first step to get the potential stops without the follow-up operation. From the table, k-medoids, DJ-Cluster and CB-SMoT are more time-consuming than the other two algorithms due to their complex computation. The efficiency of the TDBC is similar to the Time-Based and both have high clustering efficiency. In terms of precision, the TDBC and DJ-Cluster are over 0.8, which is significantly larger than the other three algorithms. In terms of recall, compared with the others, the TDBC is slightly improved. Therefore, with the promise of ensuring time efficiency, the performance of TDBC was remarkably improved.

Table 2. Results of different cluster algorithms for Dataset 1.

	TDBC	K-Medoids	DJ-Cluster	CB-SMoT	Time-Based
Parameters	d = 60 m t = 500 s	K = 6	eps = 60 m $minPoint$ = 30	$area$ = 0.3 $time$ = 300 s	d = 60 m t = 500 s
Time Complexity	$O(n)$	$O(t(n\text{-}k)^2)$	$O(tn\log n)$	$O(n^2)$	$O(n)$
Elapsed Time (s)	0.9	198.52	194.47	81.9	0.97
Precision	0.81	0.45	0.84	0.58	0.54
Recall	0.68	0.60	0.58	0.66	0.61

The treatment of GPS signal loss and data drift greatly improves the accuracy of the algorithm. Considering the recognition of the type I stop points, the recall rate demonstrated a degree of improvement. As we can see from Table 2, the recall rate is not high as a whole. We made a return visit after the experiment and found that many volunteers did not log the recorded locations strictly. For example, volunteers may record locations before they arrive or after they leave the places, or mark a location repeatedly. These factors might affect the matching accuracy and lower the recall rates in different algorithms.

Figure 10 represents the stop point extraction results of the TDBC for various types from Dataset 1. The left graph is the proportion of different types and the right is the precision of different types. From the graphs, we can see that the proportion of type III is 0.52 and the precision is 0.88, which indicates that the TDBC has a higher recognition rate for the stop points where the signal was lost for a long

time. A high rate of accuracy means that the volunteers are more inclined to log the recorded location in the building, which accords with reality. Considering most of the volunteers are students, their stop points are more often in these buildings, such as the teaching buildings, laboratories, student dormitories, libraries, etc. The proportion of type II, which is clustered by the consecutive GPS points, is 0.42, and the precision is 0.79. That indicates the GPS device has received the signal with short-range data drift in the partly closed building, which is in accordance with the raw data. Both the proportion of type I and the precision are not high, which means different volunteers had different touring habits and logged relatively few locations at the initial position of a trajectory.

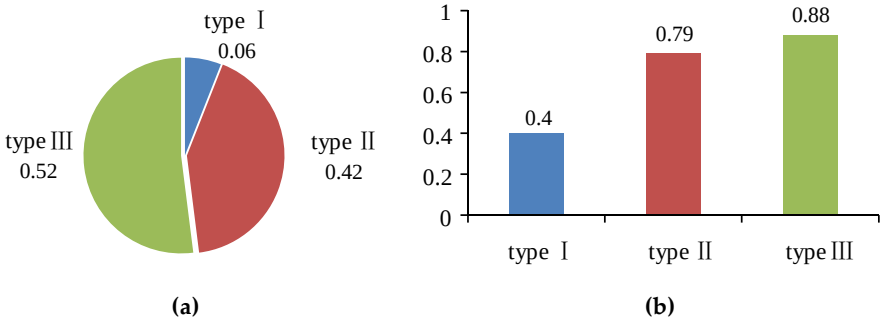

Figure 10. Details of different cluster types. (**a**) The proportion of different types; (**b**) The precision of different types.

Table 3 is the second experiment's result from Dataset 3, which only contains a day's trajectory data from one volunteer. We know the volunteer went for an outing on that day and recorded nine locations. The parameters of the other algorithms are set to the same value as the first experiment except that K-Medoids' is set to 9.

Table 3. Results of different cluster algorithms for Dataset 3.

	TDBC	K-Medoids	DJ-Cluster	CB-SMoT	Time-Based
Parameters	d =60 m t = 500 s	K = 9	*eps* = 60 m *minPoint* = 30	*area* = 0.3 *time* = 300 s	d = 60 m t = 500 s
Recorded	9	9	9	9	9
Extracted	10	9	6	11	12
Matched	7	4	5	6	6
Precision	0.7	0.44	0.83	0.55	0.5
Recall	0.78	0.44	0.56	0.66	0.66

As shown in Table 3, the clustering results of different algorithms are generally in accordance with the results of the first experiment. The precision of the DJ-Cluster and the recall of the TDBC are the highest.

The results of the comparison between the TDBC and the four algorithms are shown in Figure 11. There are five recorded locations in the visible area. The TDBC correctly found the five points and extracted the other two points, which were not recorded by the volunteer; the points A and E in the Figure 11a. The analysis shows that point A is a ferry, and that the volunteer stopped over there and waited to pass through the Yangtze River. The point E is a stop in the snack street. The volunteer had not logged either of the two points. Thus, the TDBC is reasonable. The point B in Figure 11b is a stop which experienced signal loss for a long time. The point C in Figure 11d is a stop which has points with short range data drift, and point D in Figure 11d is a circular stop.

Figure 11a is the result of K-Medoids, which accurately extracted three stop points. As the algorithm is sensitive to the initial points, different initial points may have different results. In this paper, we have experimented with the main times for the different random initial points. Some points,

e.g., the point F in Figure 11a, were still unavoidable. Obviously, these points are not reasonable stop points. Meanwhile, the time is the important characteristic of the trajectory data and it is not taken into account by the K-Medoids. Considering the time complexity and precision, the performance of the TDBC is, of course, better than the K-Medoids.

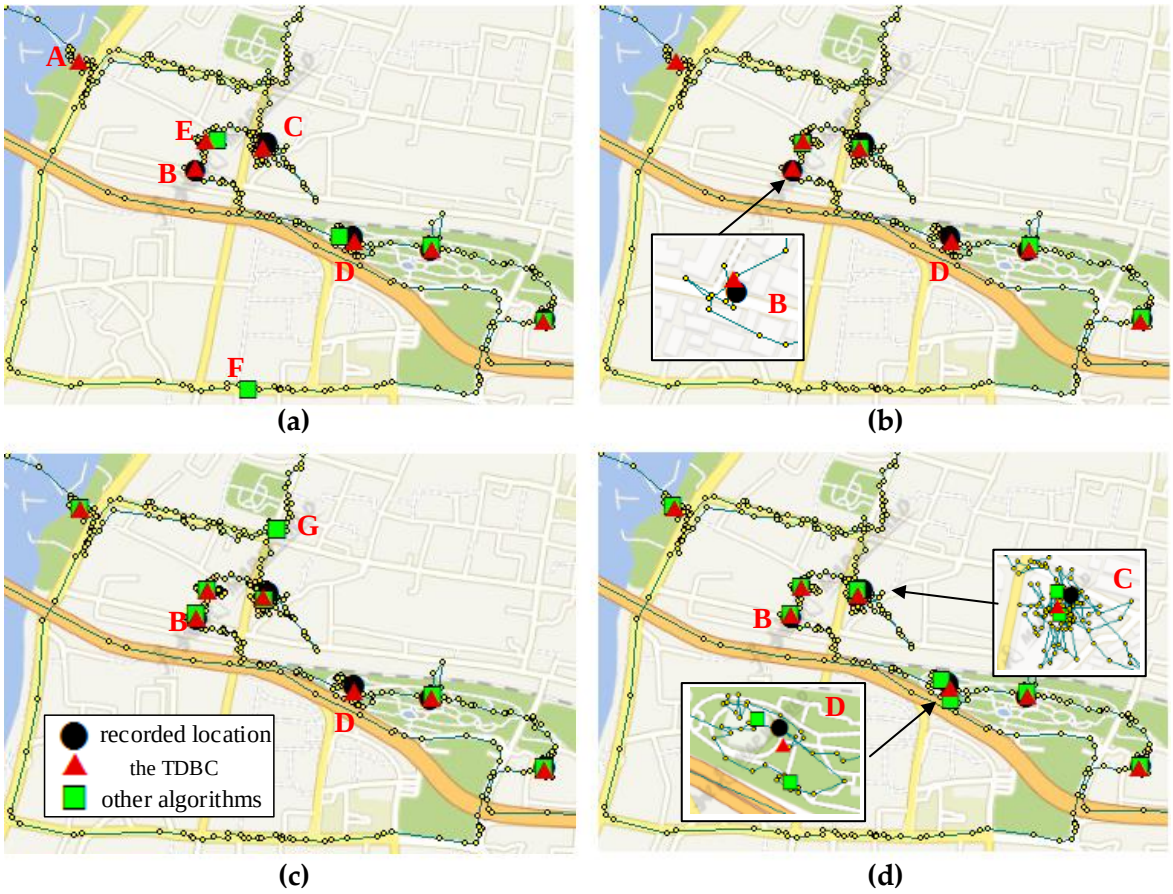

Figure 11. Results compared with other algorithms. (**a**) K-Medoids; (**b**) DJ-Cluster; (**c**) CB-SMoT; (**d**) Time-Based clustering.

Figure 11b is the result of the DJ-Cluster. The found stop points are the dense neighborhoods of GPS points, and the volunteer solely logged in these places. Since the essence of the DJ-Cluster is a density-based clustering algorithm, for those GPS points which are recorded within the same area at different times, the DJ-Cluster merges them together into a cluster and takes the cluster as a stop point without considering the influence of time. Although the results are not reasonable, the accuracy of DJ-Cluster is very high if we do not take the time into account. At the same time, the algorithm has natural defects for these stop points, which contain signal loss for an extended period, e.g., the point B in Figure 11b, and have a circular trajectory with less density, e.g., the point D in Figure 11b. That is the reason why the recall is small.

Figure 11c is the result of the CB-SMoT. Considering the influence of time, the essence of the CB-SMoT is based on the average speed to find the stop point. Therefore, the algorithm can recognize these stop points which stay outside for a long time or have signal loss, e.g., the point B in Figure 11c. It is precisely because of this characteristic that the algorithm is easy to consider in road junctions or traffic jams as stop points, e.g., the point G in Figure 11c. Also, there is no capability for a circular stop point. Therefore, the accuracy of the CB-SMoT is not high.

Figure 11d shows the time-based results, which have a higher precision. Since the algorithm considers the influence of time and has corresponding treatment measures for signal loss, these stop

points which have signal loss are correctly identified, e.g., the point B in Figure 11d. However, the algorithm divides these places which have data drift or a circular region into several stop points, such as the points C and D in Figure 11d. Both points are divided into two parts. That can lead to an increase in the total number of stop points, and decrease the accuracy of the time-based results. Meanwhile, the recall of the time-based clustering algorithm is limited because the segmentation makes more small clusters, which may not conform to the requirements of the stop point, and these clusters are ignored.

3.5. Location Extracting Experiment

The improved clustering by a fast search and identification of density peaks in this research clustered the stop points into locations from Dataset 1. Compared with the algorithms in the literature [15,17], the density-based clustering has better performance, which was proposed in the literature [3]. Therefore, our improved algorithm will be compared with the density-based one.

As shown in Figure 12, the ***Correct*** means each of the stop point matches well with the recorded location in the same group. The ***False*** means the opposite. The ***Merged*** represents whether or not stop points are matched with the recorded locations from different groups. The ***Lost*** stands for the recorded locations from the same group that do not have matched points. The ***Divided*** stands for the recorded locations that are matched with the stop points from different clusters. The precision is the proportion of the ***Correct*** clusters and the recall is the proportion of the ***Correct*** groups. The evaluation indexes include the elapsed time of the algorithm, the false rate (*FR*), merged rate (*MR*), lost rate (*LR*), divided rate (*DR*), precision (*P*), recall (*R*) and F1-Measure.

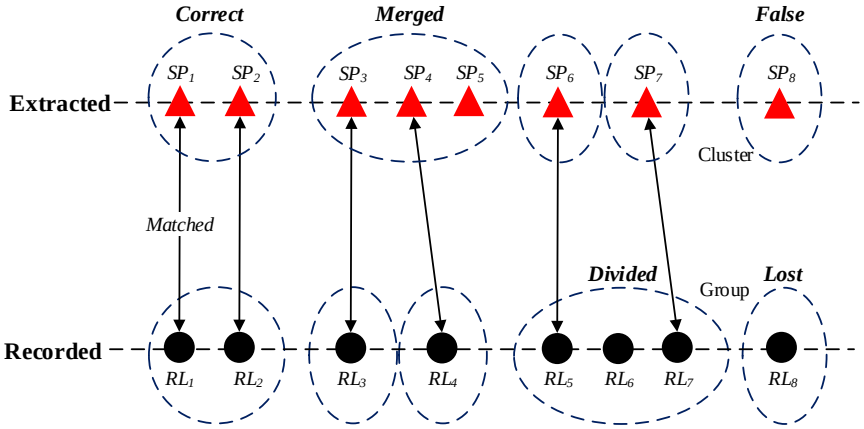

Figure 12. An example of evaluation metrics.

Table 4. Comparison with the density-based algorithm.

	Our Algorithm	Density-Based
Parameters	d = 60 m	eps = 60 m
Elapsed Time (s)	0.351	1.912
FR	0.084	0.084
MR	0.056	0.167
LR	0.163	0.186
DR	0.116	0.07
P	0.722	0.694
R	0.605	0.618
F	0.658	0.654

Table 4 is the comparison between our algorithm and the density-based clustering. The parameters of both the algorithms are set to 60 m. Compared with the density-based method, our improved

algorithm has distinct advantages in efficiency, and it has a lower ***Merged*** rate and a higher ***Divided*** rate. That is because our improved algorithm does not merge the locations which are close to each other, while the density-based one does. As shown in Figure 13, the density-based approach does not recognize the locations B and C, but merges them together. This is why the precision of our algorithm is higher. Furthermore, the false rate, lost rate and recall in the two algorithms are similar to each other. Both of them have a higher lost rate and a lower recall. This is because the overall recall rate of the algorithms is low as we discussed in Section 3.4. From the whole perspective, under the prerequisite of ensuring precision and recall rate, our improved algorithm has higher performance and efficiency.

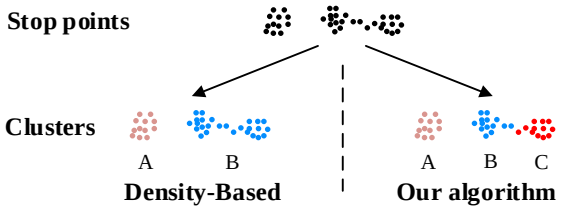

Figure 13. An example of clustering for the two algorithms.

4. Conclusions and Future Work

In this paper, we proposed a two-step clustering approach to extract personal locations from an individual GPS trajectory data. First, the three types of stop points are defined and extracted by the algorithm TDBC. Then, the improved clustering method that ran a fast search and found density peaks was used to extract locations from these stop points. Extensive experiments were performed to verify the algorithm's performance; the results demonstrate the efficiency of the proposed approach. Obviously, in terms of efficiency and performance, TDBC outperforms the existing stop points' extraction algorithm. Our new approach can substantially reduce the influence of GPS signal loss and data drift and recognize unique locations, such as circular regions. Meanwhile, the improved clustering by a fast search and identification of density peaks was applied to discover trajectory locations. This approach shows better performance than existing location clustering algorithms.

However, the proposed approach requires some prior parameters. In this paper, the parameters were prepared for the extraction of building level locations. Although the method that was used to find the optimal parameters was given, it was not automatic. In our proposed approach, manual intervention is necessary when the cluster centers are selected in Step 3. Meanwhile, analyzing the semantic information and identifying personal habits for these locations constitute the foundational work for path and location prediction, social friend recommendation, customized personalized location services, etc. These topics will be our focus in future studies.

Acknowledgments: Thank numerous individuals participated in the data acquisition.

Author Contributions: Zhongliang Fu and Zongshun Tian conceived and designed the experiments; Zongshun Tian and Changjian Qiao performed the experiments; Yanqing Xu analyzed the data; Zongshun Tian wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. González, M.C.; Hidalgo, C.A.; Barabási, A.L. Understanding individual human mobility patterns. *Nature* **2008**, *453*, 779–782. [CrossRef] [PubMed]
2. Harrison, S.; Dourish, P. Re-place-ing space: The roles of place and space in collaborative systems. In Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work, Boston, MA, USA, 16–20 November 1996.

3. Lv, M.; Chen, L.; Xu, Z.; Li, Y.; Chen, G. The discovery of personally semantic places based on trajectory data mining. *Neurocomputing* **2016**, *173*, 1142–1153. [CrossRef]
4. Yu, Y.; Kim, J.; Shin, K.; Jo, G.S. Recommendation system using location-based ontology on wireless internet: An example of collective intelligence by using 'mashup'applications. *Expert Syst. Appl.* **2009**, *36*, 11675–11681. [CrossRef]
5. Stopher, P.R.; Jiang, Q.; FitzGerald, C. Processing GPS data from travel surveys. In Proceedings of the Second International Colloqium on the Behavioural Foundations of Integrated Land-Use and Transportation Models: Frameworks, Models and Applications, Toronto, ON, Canada, 12–15 June 2005.
6. Chen, L.; Lv, M.; Chen, G. A system for destination and future route prediction based on trajectory mining. *Pervasive Mob. Comput.* **2010**, *6*, 657–676. [CrossRef]
7. Zhang, J.; Zhu, M.; Papadias, D.; Tao, Y.; Lee, D.L. Location-based spatial queries. In Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, CA, USA, 10–12 June 2003.
8. Kim, D.H.; Hightower, J.; Govindan, R.; Estrin, D. Discovering semantically meaningful places from pervasive rf-beacons. In Proceedings of the 11th international conference on Ubiquitous Computing, Orlando, FL, USA, 30 September–3 October 2009.
9. Hightower, J.; Consolvo, S.; LaMarca, A.; Smith, I.; Hughes, J. Learning and recognizing the places we go. In Proceedings of the Ubicomp 2005: Ubiquitous Computing, Tokyo, Japan, 11–14 September 2005.
10. Paek, J.; Kim, J.; Govindan, R. Energy-efficient rate-adaptive GPS-based positioning for smartphones. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, USA, 15–18 June 2010.
11. Schuessler, N.; Axhausen, K. Processing raw data from global positioning systems without additional information. *Transp. Res. Rec. J. Transp. Res. Board* **2009**. [CrossRef]
12. Ashbrook, D.; Starner, T. Learning significant locations and predicting user movement with GPS. In Proceedings of the Sixth International Symposium, Vienna, Austria, 21–23 April 2002.
13. Zhou, C.; Frankowski, D.; Ludford, P.; Shekhar, S.; Terveen, L. Discovering personally meaningful places: An interactive clustering approach. *ACM Trans. Inf. Syst. (TOIS)* **2007**. [CrossRef]
14. 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 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996.
15. Ashbrook, D.; Starner, T. Using GPS to learn significant locations and predict movement across multiple users. *Pers. Ubiquitous Comput.* **2003**, *7*, 275–286. [CrossRef]
16. Du, J.; Aultman-Hall, L. Increasing the accuracy of trip rate information from passive multi-day GPS travel datasets: Automatic trip end identification issues. *Transp. Res. Part A Policy Pract.* **2007**, *41*, 220–232. [CrossRef]
17. Kang, J.H.; Welbourne, W.; Stewart, B.; Borriello, G. Extracting places from traces of locations. In Proceedings of the 2nd ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots, Philadelphia, PA, USA, 1 October 2004.
18. Alvares, L.O.; Bogorny, V.; Kuijpers, B.; de Macedo, J.A.F.; Moelans, B.; Vaisman, A. A model for enriching trajectories with semantic geographical information. In Proceedings of the 15th annual ACM International Symposium on Advances in Geographic Information Systems, Seattle, WA, USA, 7–9 November 2007.
19. Palma, A.T.; Bogorny, V.; Kuijpers, B.; Alvares, L.O. A clustering-based approach for discovering interesting places in trajectories. In Proceedings of the 2008 ACM Symposium on Applied Computing, Ceará, Brazil, 16–20 March 2008.
20. Yuan, H.; Qian, Y.; Yang, R. Research on GPS-trajectory-based personalization poi and path mining. *Syst. Eng. Theory Pract.* **2015**, *35*, 1276–1282.
21. Zaïane, O.R.; Foss, A.; Lee, C.H.; Wang, W. On data clustering analysis: Scalability, constraints, and validation. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Taipei, Taiwan, 6–8 May 2002.
22. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [CrossRef] [PubMed]