A Historical-Trajectories-Based Map Matching Algorithm for Container Positioning and Tracking

Wenfeng Li, Wenwen Zhang and Cong Gao *

School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China; liwf@whut.edu.cn (W.L.); zww96@whut.edu.cn (W.Z.)
* Correspondence: gaoc@whut.edu.cn

Abstract: Positioning and tracking of containers is becoming an urgent demand of container transportation. Map matching algorithms have been widely applied to correct positioning errors. Because container trajectories have the characteristics of low sampling rate and missing GPS points, existing map matching algorithms based on the shortest path principle are not applicable for container positioning and tracking. To solve this problem, a historical-trajectories-based map matching algorithm (HTMM) is proposed. HTMM mines the travel time and the frequency in historical trajectories to help find the local path between two adjacent candidate points. HTMM first presents a path reconstruction method to calculate the travel time of historical trajectories on each road segment. A historical path index library based on a path tree is then developed to efficiently index historical paths. In addition, a location query and tracking method is introduced to determine the location of containers at given time. The performance of HTMM is validated on a real freight trajectory dataset. The experimental results show that HTMM has more than 3% and 5% improvement over the ST-Matching algorithm and HMM-based algorithm, respectively, at 60–300 s sampling intervals. The positioning error is reduced by half at a 60 s sampling interval.

Keywords: container positioning and tracking; map matching; historical trajectories; path reconstruction

1. Introduction

As the core of cargo transportation, containers need to be positioned and tracked remotely in transit during transportation. The positioning and tracking of containers is mostly realized by installing satellite positioning devices on containers, collecting and transmitting data to a remote monitoring platform [1]. Satellite positioning devices are usually powered by microbatteries. Considering the high energy consumption of positioning and communication and the long transportation cycle, the sampling rate is generally set at 2 min or more to extend the battery life [2]. The distance between two adjacent GPS points is too far and cannot accurately reflect the real path of a container [3]. Moreover, the positioning signal becomes weak or even interrupted in tunnels or under viaducts and tall buildings, which causes missing and inaccurate trajectories [4]. These problems cause blind zones in the positioning and tracking of containers, and high precision positioning cannot be achieved. Therefore, accurate positioning under scenarios of low sampling rate and missing and inaccurate trajectories is a major challenge in container transportation.

Map matching algorithms, which correct errors by matching GPS points to the most probable road, have been verified as effective for map matching of passenger transport system [5,6]. To date, there are few map matching algorithms for container transportation with low-sampling-rate trajectories. Hidden Markov model (HMM)-based map matching algorithms (such as ST-Matching algorithm [7]) have been applied to match trajectories with low sampling rates of 60–300 s [8]. The commonality of these algorithms is that they choose the shortest path between two adjacent candidate points as the local path [9]. Drivers do not choose the shortest path in many cases, which reduces the accuracy of
map matching algorithms. On the one hand, container transport is accompanied by cargo handling, vehicle maintenance, high toll fees and loan guarantee, etc. [10]. Drivers are more inclined to choose familiar paths or paths with relatively low comprehensive cost rather than the shortest path. On the other hand, with the decrease in sampling rate, the number of potential matched paths between two adjacent GPS points increases. Therefore, using a shortest-path-based map matching algorithm to match low-sampling-rate trajectories of container transport can easily lead to matching errors.

The information in historical trajectories, including travel time, frequency and driving preference of drivers, is not fully utilized in current map matching algorithms. Based on this, we wonder whether potential information can be mined from historical trajectories to help map matching of container transport with low sampling rates and missing GPS points. To implement this idea, a novel historical-trajectories-based map matching algorithm (HTMM) is presented in this paper. HTMM is an improved HMM-based map matching algorithm. In HTMM, information from historical trajectories, including travel time and frequency, is mined to help determine the local path. The historical velocity, distance and heading angle are employed in the search for an optimal path. Experiments on a real freight trajectory dataset verify the effectiveness of HTMM. In general, the major contributions of this paper are as follows:

1. A historical-trajectories-based map matching algorithm (HTMM) consisting of local candidate path selection and optimal matched path search is proposed. The local path selection utilizes the travel time and the frequency mined from historical trajectories. Historical speed is introduced to improve the transition probability in the optimal matched path search.

2. A historical path reconstruction and index method is proposed to make full use of the historical paths. A meta path is introduced to determine the travel time of historical paths. A historical path index library based on a path tree is created to store historical information and accelerate the index speed.

3. A location correction and query method is presented to reconstruct matched paths and calculate the location of containers. Through this method, location correction and continuous tracking of the container are realized.

The outline of this paper is as follows. Section 2 reviews the related work of container positioning and tracking technology and map matching algorithms. Section 3 introduces a detailed discussion on the proposed HTMM algorithm. Section 4 verifies the accuracy and effectiveness of HTMM through experiments and displays the experimental results. Section 5 gives the conclusions and discusses future work.

2. Related Work

The Internet of Things (IoT) endows containers with intelligence, which is the basis of container positioning and tracking [11]. A container equipped with an IoT system collects identity, location, temperature and other data and transmits these data to a remote monitoring platform through a cellular mobile network, low-power LAN, LEO satellite, etc. The remote monitoring platform stores and process these data and provides data services for specified scenarios, among which container positioning and tracking is most widely demanded. For this purpose, many researchers have put forward different solutions. Nie et al. [12] designed an intelligent terminal system based on LEO satellite and narrowband Internet of Things (NB-IoT) that solves the problems of high power consumption and insufficient coverage of terminals. Lu et al. [13] proposed a container multimodal transport platform based on space-based IoT that can provide container tracking and monitoring services for different carries, such as highway, railway and waterway transportation. Tang et al. [14] proposed a refrigerated container monitoring system based on a wireless sensor network in which a geographic information system is used. This system can display container location, time, temperature and humidity information on Google Maps.

In summary, the mentioned literature points out that the following two problems need to be addressed in the design and development of positioning terminals: (1) low cost
and (2) low power consumption. These two problems lead to low positioning precision and incomplete tracking of containers. To date, few researchers have come up with useful solutions.

One method to improve positioning precision is to use pseudo-range differential positioning devices with sub-meter precision; however, high cost, missing trajectories and drift cannot be avoided. Map matching technology is a good choice to improve precision and reduce cost. Besides improvement of positioning precision, on a deeper level, trajectories after map matching can be applied to advanced applications, such as logistics tracking, location prediction and path planning or recommendation [15]. Missing and low sampling rate (60 s or more) trajectories are the research object of this paper. Simple map matching algorithms, such as geometry-based [16] and topology-based [17] map matching algorithms, have higher accuracy when the sampling rate is 1–10 s but perform worse when matching low-sampling-rate trajectories.

There are two types of map matching algorithms suitable for low-sampling-rate trajectories. One is map matching based on HMM. Newson et al. [8] first applied HMM to map matching; the accuracy can reach 99% when the sampling rate is 1–30 s but gradually decreases with sampling rates over 30 s. This map matching algorithm considers various features, such as distance and direction, when calculating observation and transition probabilities. Most assume these features satisfy a certain probability distribution. Lou et al. [7] proposed the ST-Matching algorithm on this basis. The ST-Matching combines spatial and temporal features; in particular, the road speed limit is introduced into the temporal feature. The accuracy of this algorithm can reach 80% when the sampling rate is 2.5 min. Yuan et al. [9] proposed the IVMM algorithm, which considers the influence of the context GPS point to match the current GPS point. It adds an interacting voting process on the basis of ST-Matching. When the sampling rate is 1.5–6.5 min, the accuracy of IVMM is about 70%. There are many similar map matching algorithms, such as those presented in [18,19].

Map matching algorithms based on HMM assume that the local candidate path between two adjacent points follows the shortest length/time path, which is not suitable for all low-sampling-rate trajectories and dense road networks. Map matching algorithms based on local path inference that do not rely on shortest length/time path assumptions have been proposed to solve this problem. Zheng et al. [20] believe that the user’s trajectories follow certain space–time travel rules and proposed a two-stage path inference system, HRIS, driven by historical data. The first stage is local path inference, which calculates multiple possible paths between any two adjacent points. In the second stage, a dynamic programming algorithm is used to calculate the most probable path from the local paths as the optimal matched path. Banerjee et al. [21] proposed a path inference system, InferTra, that infers the complete path of a trajectory using the Gibbs sampling algorithm and the network mobility model. Wu et al. [22] proposed a completely probabilistic path inference system, STRS, that also considers the temporal and spatial probability distribution models of historical trajectories. Compared with other map matching algorithms, the accuracy of STRS is greatly improved.

In essence, these algorithms solve the most probable path through the assumed probability distribution or prior historical trajectories. However, there are some problems in solving the positioning and tracking of container transportation:

(1) Map matching algorithms based on HMM follow the shortest length/time principle, which is not applicable for missing and low-sampling-rate trajectories. Map matching algorithms based on local path inference require a large number of historical trajectories to extract path selection preference and travel law, which lead to high time complexity.

(2) With the rapid development of roads and traffic control, only using movement modes found from historical trajectories may not be suitable for the current situation.
(3) In passenger transportation systems, trajectories come from multiple drivers with different purposes, and the movement modes found from all historical trajectories cannot represent each driver’s driving preferences.

(4) Most of the above algorithms focus on passenger transportation systems, especially floating cars in an urban road network. Container transportation systems are rarely studied, but the demand for positioning and tracking of containers is considerable.

Motivated by these problems, a map matching algorithm called HTMM is developed in this paper. The main differences between this work and previous work are as follows:

(1) HTMM is a two-stage map matching algorithm based on a hidden Markov model and a local path inference method.

(2) Both historical information and real-time vehicle movement information are considered in HTMM.

(3) HTMM aims to improve positioning precision and track the location of containers in a container transportation system.

3. Methodology
3.1. Preliminary

In this subsection, some definitions of the map matching problem will be given as follows.

Definition 1 (Trajectory). Trajectory, $T$, is defined as a sequence of GPS points with timestamps, i.e., $T = \langle p_1, p_2, \ldots, p_n \rangle$. A GPS point is expressed as $p_i = \langle \text{lat}, \text{lng}, t, v, \gamma \rangle$ ($1 \leq i \leq n$), which consists of the latitude ($p_i$.lat), the longitude ($p_i$.lng), the velocity ($p_i$.v) and the heading angle ($p_i$.gamma) of $p_i$ at time $p_i.t$.

Definition 2 (Road Network). Road network is defined as a directed graph, $G = \{V, E\}$. Node set $V$ represents the set of end or intersection of road segments. Edge set $E$ represents the set of road segments that connect the nodes in $V$. The $e_i (1 \leq j \leq |E|)$ in $E$ has three attributes: (1) the unique number of $e_i$ ($e_i$.eid); (2) the coordinates of the start node ($e_i$.start) and the end node ($e_i$.end) of $e_i$; and (3) the length of $e_i$ ($e_i$.length). $|E|$ is the number of edges in $E$.

Definition 3 (Candidate road segment and candidate point). Given $p_i$ and $G = \{V, E\}$, the candidate road segments of $p_i$ are defined as $E_i = \{e_j | d(p_i, e_j) < \delta, \forall e_j \in E\}$, where $d(p_i, e_j)$ is the projection distance from $p_i$ to $e_j$, and $\delta$ is the distance threshold. The candidate point of $p_i$ on $e_j$ is expressed as $c_j^i$.

Definition 4 (Path). A path is a sequence of road segments. As shown in Figure 1, the path from $c_i^1$ to $c_i^{n+1}$ is expressed as $P_{r \rightarrow s} : \{e_r \rightarrow e_{r+1} \rightarrow \cdots \rightarrow e_s\}$, in which $c_i^1$ is on $e_r$ and $c_i^{n+1}$ is on $e_s$.

Figure 1. The path from the candidate point, $c_i^r$ on $e_r$ to the candidate point $c_i^{r+1}$ on $e_s$. $c_i^r$ is the projection point of $p_i$, and $c_i^{r+1}$ is the projection point of $p_{i+1}$.

Definition 5 (Matched point). The candidate point on the global optimal path is the matched point. The matched point of GPS point $p_i$ on road segment $e_j$ is expressed as $m_i^j$. $m_i^j$ has two attributes: (1) the error of the GPS point and the matched point ($m_i^j$.error); (2) the distance from $m_i^j$ to the start node of $e_j$ ($m_i^j$.offset set).
Due to the measurement error of the positioning device, the GPS points are not always on the road of the map. Therefore, the goal of the map matching algorithm is to find the most probable path under the given trajectory, $T$, and the road network, $G = \{V, E\}$.

### 3.2. Notation Description

Table 1 presents the relevant indices and variables for HTMM.

| Type       | Name                  | Meaning                                      |
|------------|-----------------------|----------------------------------------------|
| Indices    | $i$                   | Index of GPS points                          |
|            | $j, r, s$             | Index of edges                               |
|            | $k$                   | Index of candidate path                      |
|            | $p_i$                 | The $i$th GPS point                          |
|            | $p_i\.lat, p_i\.lng$ | The latitude and longitude of the $i$th GPS point |
|            | $p_i\.t$              | The timestamp of the $i$th GPS point          |
|            | $p_i\.v$              | The velocity of the $i$th GPS point           |
|            | $p_i\.\gamma$        | The heading angle of the $i$th GPS point      |
|            | $e_j$                 | The $j$th edge                               |
|            | $e_j\.eid$            | The number of $e_j$                          |
|            | $e_j\.start, e_j\.end$ | The start and end node of $e_j$              |
|            | $e_j\.length$        | The length of $e_j$                          |
|            | $c_i$                 | The candidate point of $p_i$ on $e_j$         |
|            | $P_{r \rightarrow s}$ | The path from $e_r$ to $e_s$                 |
|            | $m_i$                 | The matched point of $p_i$                   |
|            | $m_i\.error$         | The distance from $p_i$ to $m_i$              |
|            | $m_i\.offset$        | The distance from $m_i$ to $e_i\.start$      |
| Variables  | $MP^k_{j, enter\_time(leave\_time)}$ | The enter time (or leave time) of $MP^k_j$ |
|            | $l$                   | The number of candidate paths                |
|            | $CP_k$                | The $k$th candidate path                     |
|            | $T_k$                 | The $k$th trajectory                         |
|            | $t_j$                 | The travel time of $T_k$ on $e_j$            |
|            | $\bar{t}_j$          | The estimated travel time of $e_j$           |
|            | $s^k_{time}$          | The similarity of $CP_k$ on travel time       |
|            | $c_i^k$               | The number of times of $CP_k$ on $e_j$        |
|            | $F_k$                 | The frequency of $CP_k$                      |
|            | $S_k$                 | The total similarity of $CP_k$               |
|            | $p_{r}(c_i^k)$        | The observation probability of $c_i^k$       |
|            | $\bar{\tau}_{r \rightarrow s}$ | The average speed from $p_i$ to $p_{i+1}$ |
|            | $v_j$                 | The historical speed of $e_j$                |
|            | $p_l(c_i^r \rightarrow c_{i+1}^r)$ | The transition probability from $c_i^r$ to $c_{i+1}^r$ |

### 3.3. System Overview

As shown in Figure 2, the framework of HTMM consists of three parts: the historical path reconstruction and index module, the local candidate selection module, and the location correction and query module. In the historical path reconstruction and index module, the historical trajectories with high sampling rates are matched to determine the historical paths. A path reconstruction method is proposed to generate meta paths, which record the enter time and the leave time of the historical paths. A historical path index library is built to index paths efficiently. In the local candidate path selection module, a local candidate path selection strategy is introduced to determine the path between two adjacent candidate points, in which the similarity of travel time and the frequency are calculated. In the location correction and query module, an improved matched point search method based on the HMM model is used to calculate the matched points. A location query
3.3. System Overview

As shown in Figure 2, the framework of HTMM consists of three parts: the historical path reconstruction and index module, a local candidate path selection module, and a location correction and query module.

3.4. Historical Path Reconstruction and Index Module

In this subsection, a historical path reconstruction and index method is proposed to make better use of the historical paths.

3.4.1. Map Matching

The map matching algorithm based on HMM proposed by Newson et al. [8] is utilized to obtain the real paths of the historical trajectories. This algorithm has 99% accuracy when the sampling rate is under 30 s. The matched paths can be considered to represent where the containers really pass through.

3.4.2. Path Reconstruction

After map matching, two adjacent matched points may be on the same road segment or across multiple road segments, so the travel time of historical paths on each road segment are unknown. A path reconstruction method based on time and distance interpolation is proposed in this subsection. A constructed path consists of several meta paths, which are defined as follows:

**Definition 6 (Meta Path).** The meta path of road segment \( e_j \) is represented as \( MP_j \), which is associated with the enter time on \( e_j \) (\( MP_j.\text{enter\_time} \)) and the leave time on \( e_j \) (\( MP_j.\text{leave\_time} \)). The path \( P_{r\rightarrow s} = \{ e_r \rightarrow e_{r+1} \rightarrow \cdots \rightarrow e_s \} \) consists of a meta path sequence \( \{ MP_r, MP_{r+1}, \cdots, MP_s \} \).

The travel time of historical paths is calculated to reconstruct the meta path. Three situations of the location relationship of two adjacent matched points need to be considered. **Situation 1:** \( m_r \) and \( m_{r+1} \) are on the adjacent road segments \( e_r \) and \( e_{r+1} \), respectively.
As shown in Figure 3a, the leave time of \( MP_r \) is equal to the enter time of \( MP_{r+1} \), i.e., \( MP_r.\text{leave}_\text{time} = MP_{r+1}.\text{enter}_\text{time} \). It is supposed that the carrier of the container travels at a constant speed. \( MP_r.\text{leave}_\text{time} \) is calculated as:

\[
\begin{align*}
MP_r.\text{enter}_\text{time} &= p_i.t + (p_{i+1}.t - p_i.t) \times \frac{\text{dist}(m'_i, e_r.\text{end})}{\text{len}(m'_i, m'_{i+1})} \\
\text{dist}(m'_i, e_r.\text{end}) &= e_r.\text{length} - m'_i.\text{offset} \\
\text{len}(m'_i, m'_{i+1}) &= \text{dist}(m'_i, e_r.\text{end}) + m'_{i+1}.\text{offset}
\end{align*}
\]

(1)

where \( \text{dist}(m'_i, e_r.\text{end}) \) is the distance between \( m'_i \) and \( e_r.\text{end} \), and \( \text{len}(m'_i, m'_{i+1}) \) is the path length between \( m'_i \) and \( m'_{i+1} \).

Figure 3. Three situations of the location relationship of adjacent two matched points: (a) \( m'_i \) and \( m'_{i+1} \) are on adjacent road segments; (b) \( m'_i \) and \( m'_{i+1} \) are separated by multiple road segments; (c) \( m'_i \) and \( m'_{i+1} \) are on the same road segment.

**Situation 2**: \( m'_i \) and \( m'_{i+1} \) are separated by multiple road segments.

As shown in Figure 3b, other road segments between \( e_r \) and \( e_r \) have no matched points. The enter time and leave time of the meta paths of these road segments are calculated by time and distance interpolation, as formulated in Equation (2):

\[
\begin{align*}
\text{MP}_r.\text{leave}_\text{time} &= p_i.t + (p_{i+1}.t - p_i.t) \times \frac{\text{dist}(m'_i, e_r.\text{end})}{\text{len}(m'_i, m'_{i+1})} \\
\text{MP}_{r+1}.\text{leave}_\text{time} &= \text{MP}_r.\text{leave}_\text{time} + (p_{i+1}.t - p_i.t) \times \frac{e_r.\text{length}}{e_r.\text{length}} \\
\text{MP}_s.\text{leave}_\text{time} &= \text{MP}_{s-1}.\text{leave}_\text{time} + (p_{i+1}.t - p_i.t) \times \frac{e_s.\text{length}}{e_s.\text{length}}
\end{align*}
\]

(2)

**Situation 3**: \( m'_i \) and \( m'_{i+1} \) are on the same road segment, \( e_r \).

As shown in Figure 3c, the matched point, \( m'_i \), does not take new information for the calculation of its travel time. The enter time and the leave time of \( MP_r \) should be calculated according to the next matched point that is not on \( e_r \).

For the meta path of the road segment on which the first (or last) matched point is located, the enter time (or the leave time) is equal to the timestamp of the first (or the last) GPS point.

3.4.3. Historical Path Indexing

To accelerate the indexing speed of the historical paths, a historical path index structure called an HP-tree is built based on a path tree. The HP-tree is an improvement on FP-tree proposed by Huang et al. [23,24]. The HP-tree has the following characteristics:

1. Each node of the HP-tree represents a road segment, among which the root node is the start road segment of the historical path (or sub path) and the leaf node is the end road segment of the historical path (or sub path).
2. The child node of a node represents the next road segment to which it directly connects.
3. All HP-trees make up an HP-forest to represent all historical paths and sub paths.
Each node in the HP-tree is associated with: (1) sid, which represents the road segment number and equals to e{id} of the node; (2) cnt, which represents the number of times the node is passed through; and (3) stime, which represents the travel time of historical paths on the node. stime is represented in the form of a dictionary, the key of which is the trajectory id, tid, the value of which is a list of travel times of historical paths on the road segment.

Each HP-tree is associated with: (1) the root node, root and (2) the relational list, idx_list. idx_list is represented in the form of a dictionary, the key of which is the road segment number, sid, the value of which is a list of nodes with the same sid.

As shown in Figure 4, there are 4 historical paths: P_{1→7} = \{e_1 \rightarrow e_2 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7\}, P_{1→5} = \{e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_5\}, P_{2→7} = \{e_2 \rightarrow e_3 \rightarrow e_5 \rightarrow e_7\} and P_{2→6} = \{e_2 \rightarrow e_4 \rightarrow e_6\}. Seven HP-trees with e_1 \rightarrow e_7 as root nodes can be created. Only the index structure of two of them are shown in Figure 5. With the help of an HP-tree, historical paths between given start and end road segments can be efficiently searched. For example, to search all the historical paths between e_2 and e_7, the HP-tree with root node e_2 is first found; then, the relational list, idx_list, with key e_7,eid is searched. Each node in idx_list is traversed, and its parent node is recursively traversed until the root node is reached. The historical paths are obtained by outputting these nodes in reverse order. Therefore, the historical paths are \{e_2 \rightarrow e_3 \rightarrow e_5 \rightarrow e_7\} and \{e_2 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7\}. cnt and stime of each road segment in historical paths can be obtained at the same time.

![Figure 4. Four trajectories in the road network: P_{1→7} = \{e_1 \rightarrow e_2 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7\} (red line), P_{1→5} = \{e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_5\} (blue line), P_{2→7} = \{e_2 \rightarrow e_3 \rightarrow e_5 \rightarrow e_7\} (green line) and P_{2→6} = \{e_2 \rightarrow e_4 \rightarrow e_6\} (yellow line).](image)

![Figure 5. Illustration of HP-forest. The HP-tree with root e_1 stores two historical paths: \{e_1 \rightarrow e_2 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7\} and \{e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_5\}. The HP-tree with root e_2 stores two historical paths: \{e_2 \rightarrow e_3 \rightarrow e_5 \rightarrow e_7\} and \{e_2 \rightarrow e_4 \rightarrow e_6\}.](image)
3.5. Local Candidate Path Selection Module

For the missing and low-sampling-rate trajectories of containers, the candidate points of two adjacent GPS points may not be in the same road segment or adjacent road segments. There may be multiple paths to select between two adjacent GPS points. It is assumed that the path between two adjacent GPS points follows the shortest path principle in the map matching algorithm based on HMM and several improved algorithms. In Figure 6a, only one path between two candidate points, \(P_1\) and \(P_2\), of high-sampling-rate trajectories is found, which is the shortest path. However, in Figure 6b, 3 possible paths are found between \(P_1\) and \(P_2\) of low-sampling-rate trajectories. Based on the shortest path principle, \(P_2\) is selected as the local candidate path. However, drivers may choose \(P_1\) or \(P_3\) under different traffic conditions and based on personal preferences.

![Possible paths from candidate point \(c_i\) to \(c_{i+1}\); (a) for high-sampling-rate trajectories, both the shortest path and the local candidate path is \(P_1\); (b) for low-sampling-rate trajectories, \(P_2\) is the shortest path, but the local candidate path may be \(P_1\), \(P_2\) or \(P_3\).](image)

Figure 6. Possible paths from candidate point \(c_i\) to \(c_{i+1}\); (a) for high-sampling-rate trajectories, both the shortest path and the local candidate path is \(P_1\); (b) for low-sampling-rate trajectories, \(P_2\) is the shortest path, but the local candidate path may be \(P_1\), \(P_2\) or \(P_3\).

A local candidate path selection strategy that takes the travel time and the frequency of historical paths into consideration is proposed in this paper and is elaborated in the following section.

3.5.1. Possible Paths Search

To find all the possible paths between two adjacent candidate points, the directed graph, \(G = (V, E)\); the end node of \(e_r\); and the start node of \(e_s\) are found and viewed as the source node and the target node, respectively. The steps of the search strategy are as follows:

1. Search the historical paths between \(e_r.end\) and \(e_s.start\) in the HP-Forest;
2. If no historical path is found, the shortest path between \(e_r.end\) and \(e_s.start\) is selected as the local candidate path;
3. If only one historical path is found, then it is the local candidate path. If multiple historical paths are found, then add all these historical paths to the possible path set.

3.5.2. Similarity Calculation

1. Travel Time Estimation

Considering road conditions and driving habits of drivers, the shortest path is not necessarily the best choice. It is assumed that the driver is more likely to choose the
path with a travel time is close to that of the current trajectory. Road-segment-based and path-based methods are commonly used for path travel time estimation [22]. The travel time of a road segment is obtained according to the estimated average speed in the road-segment-based method. The travel time of a path is the sum of the travel time of all the road segments. However, it cannot be used very effectively when speed changes suddenly at crossroads or speed is not available. In a path-based method, the average time of paths is calculated by retrieving the same paths from the historical paths. Although speed estimation is avoided, there are rarely two identical paths in the historical paths.

A travel time estimation method combining a road-segment-based method and a path-based method is proposed in this paper. Because the possible paths between two adjacent candidate points are the sub paths of the historical paths, the problem with the path-based method is avoided. The travel time estimation of road segments is obtained by calculating the average travel time of the corresponding road segment from multiple identical possible paths. The estimated travel time of the path is equal to the sum of estimated travel time of these road segments.

The possible path set between \( c_r \) and \( c_{s+1} \) is represented as \( CP = \{CP_1, CP_2, \ldots, CP_l \} \). Assume the historical trajectories that match \( CP_k = \{e_r, e_{r+1}, \ldots, e_s\} \) \((k = 1, 2, \ldots, l)\) are \( T_1, T_2, \ldots, T_l \). The travel time of \( T_k \) on each road segment is represented as \( t_{kr}, t_{kr+1}, \ldots, t_{ks} \). The travel time of the intermediate road segments, \( e_r, e_{r+1}, \ldots, e_{s-1} \), that are passed through completely is formulated as:

\[
\begin{align*}
\bar{t}_{r+1} & = \frac{1}{l} \sum_{k=1}^{l} t_{kr+1} / l \\
\bar{t}_{r+2} & = \frac{1}{l} \sum_{k=1}^{l} t_{kr+2} / l \\
\vdots \\
\bar{t}_{s-1} & = \frac{1}{l} \sum_{k=1}^{l} t_{ks-1} / l 
\end{align*}
\] (3)

For \( e_r \) and \( e_s \), the location of the matched point on the road segment needs to be considered when calculating the travel time. The calculation formula is:

\[
\begin{align*}
\bar{t}_r & = (\sum_{k=1}^{l} \frac{t_k}{T_k \cdot m_{i, off set}}) / l \\
\bar{t}_s & = (\sum_{k=1}^{l} \frac{t_k}{T_k \cdot m_{i+1, off set}}) / l 
\end{align*}
\] (4)

Therefore, the similarity of the time difference between \( c_r^e, c_{i+1}^e \) and the estimated travel time of \( CP_k \) is:

\[
S_{k time}^{time} = \frac{(p_{i+1, t} - p_i \cdot t) - \sum_{j=r}^{s} \bar{t}_j}{(p_{i+1, t} - p_i \cdot t)} 
\] (5)

(2) Frequency Calculation

Path frequency is the second consideration when selecting the local candidate path. Through the observation of drivers, Gao et al. [25] found that about 60% of driving paths are repeated for at least 40 days.

The number of times all road segments in \( CP_k \) are passed through by historical trajectories is represented as \( c_{r}^k, c_{r+1}^k, \ldots, c_{s}^k \). It is found that the number of times that \( CP_k \) is passed through is equal to \( c_{s}^k \).

Therefore, the frequency of \( CP_k \) is:

\[
F_k = \frac{c_{s}^k}{\sum_{k=1}^{l} c_{s}^k} 
\] (6)

(3) Path Selection
The travel time similarities and the frequencies of all possible paths in CP are calculated by integrating the weights, \( w_{\text{time}} \) and \( w_{\text{fre}} \). The comprehensive score, \( S_k \), is formulated as:

\[
S_k = w_{\text{time}} \times S_k^{\text{time}} + w_{\text{fre}} \times F_k
\]  

(7)

The local candidate path is the most similar with the largest \( S_k \).

3.6. Location Correction and Query Module

A matched point search strategy is proposed in this subsection. More specifically, the matched point is the corrected GPS point. Because the GPS points are discrete, the location of a GPS point cannot be known every time. Therefore, a method of location query of GPS points is introduced.

3.6.1. Optimal Matched Point Search

As shown in Figure 7, each GPS point has multiple candidate points, so there are multiple local candidate paths between two adjacent GPS points. In order to determine which candidate point the GPS point matches to, the probabilities of all local candidate paths are calculated.

(1) Observation Probability

The observation probability represents the possibility that \( p_i \) matches to \( c_i^r \). The distance and the heading angle are taken into consideration; then, the observation probability \( p_o(c_i^r) \) is formulated as:

\[
p_o(c_i^r) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{\text{dist}(p_i, c_i^r)}{2\sigma^2}} \times \lambda e^{-\lambda\theta}
\]  

(8)

where \( \text{dist}(p_i, c_i^r) \) is the distance between \( p_i \) and \( c_i^r \) and obeys the normal distribution with mean zero and variance \( \sigma \). \( \theta \) is the angle between the driving direction and the direction of the road segment and obeys exponential distribution with rate parameter \( \lambda \).

(2) Transition Probability

The transition probability represents the possibility that the next path of \( c_i^r \) is \( c_e \). In most map matching algorithms based on HMM, it is supposed that the transition probability is associated with the length difference between the length of a local candidate path and the distance between \( p_i \) and \( p_{i+1} \). As a result, the output tends to be the shorter path. In fact, the longer paths are filtered in the local candidate path selection module of the system, so only average velocity is considered.

Figure 7. Candidate graph composed of candidate points and local candidate paths. Each candidate point has an observation probability, and the probability from the previous candidate point to the current candidate point is expressed as a transition probability.
It is assumed that one of the local candidate paths from $c_r$ to $c_{s+1}$ is $\{e_r, e_{r+1}, \cdots, e_s\}$ because the travel time of the current trajectory on each road segment is unknown, and only the average speed can be calculated, as shown in Equation (9):

$$\tau_{i\to i+1} = \frac{\sum_j e_j.length}{\Delta t_{i\to i+1}}$$

(9)

where $\Delta t_{i\to i+1}$ is the time difference between $p_i$ and $p_{i+1}$.

The ST-Matching algorithm [7] calculates velocity similarity by comparing the difference between the speed limit and the average velocity. Due to the velocity change and waiting time caused by road traffic congestion, the velocity is always much lower than the speed limit. In addition, it is difficult to obtain real-time traffic flow information. Therefore, HTMM calculates velocity similarity by comparing the difference between the historical velocity and the average velocity. The historical velocity of the road segment of the local candidate path is:

$$v_j = e_j.length / t_j (j = r, r+1, \cdots, s)$$

(10)

Therefore, the transition probability is:

$$p_t (c'_r \to c'_{r+1}) = \frac{\sum_{j=r}^{s} v_j \times \tau_{j\to i+1}}{\sqrt{\sum_{j=r}^{s} (v_j)^2} \sqrt{\sum_{j=r}^{s} (\tau_{j\to i+1})^2}}$$

(11)

The optimal matched points are denoted as $\{m'_r \to m'_{r+1} \to \cdots \to m'_n\}$, which consists of the real path. From the first GPS point, the union probability is calculated recursively until the last GPS point is reached. The union probability is the product of the observation probability and the transition probability. The Viterbi algorithm is used to obtain the solution.

3.6.2. Location Query and Tracking

In order to achieve more accurate query of cargo shippers and carriers on the locations of containers, a container location query and tracking method based on matching results is proposed.

Given time $t$, the location ($p'$) of the container is obtained with the following steps:

1. Reconstruct the matched path as meta paths to obtain the enter time and the leave time according to the method in Section 3.4.2. The meta path is represented as $(MP_1, MP_2, \cdots)$.
2. Find the meta path $(MP_j)$ where the container is located, which satisfies:

$$MP_j.\text{enter\_time} \leq p'.t \leq MP_j.\text{leave\_time}$$

(12)

so that the container is on $e_j$.

3. Obtain the latitude and longitude of $p'$ according to the interpolation of time and average velocity, as shown in Equation (13):

$$\begin{align*}
p'.\text{lat} &= e_j.\text{start\_lat} + \frac{|t-MP_j.\text{enter\_time}| \times \sigma}{e_j.length} \times (e_j.\text{end\_lat} - e_j.\text{start\_lat}) \\
p'.\text{lng} &= e_j.\text{start\_lng} + \frac{|t-MP_j.\text{enter\_time}| \times \sigma}{e_j.length} \times (e_j.\text{end\_lng} - e_j.\text{start\_lng})
\end{align*}$$

(13)

where $\sigma$ represents the average velocity of the GPS point.

The reconstructed path can be added to the historical index library again, which reinforces the positioning and tracking of later GPS points.
4. Experiments

In this section, the effectiveness of HTMM is verified on a real trajectory dataset. HTMM is compared with two baseline algorithms by evaluating the matching accuracy and the matching time. The matched results in two typical situations are visualized to illustrate the improvement of HTMM. The positioning errors before and after location correction are calculated to show the effect of map matching in reducing positioning error.

4.1. Trajectory Dataset and Road Network

The experimental dataset comes from the real-time GPS positioning records of 15000 trucks of a freight company in Shenzhen and surrounding cities. The places of departure and destination include container terminals, logistics parks, ports, etc., as shown in Figure 8.

![Figure 8. Distribution of the GPS trajectories. The green background is the road network, and other colored lines are trajectories.](image)

HTMM is performed on a region whose latitude and longitude range is (22.5424, 113.8489, 22.5966, 113.9257). The road network downloaded from Open Street Map is contains 17,108 nodes and 18,892 edges, as shown in Figure 9.

![Figure 9. Road network of the region. The network contains 17,108 nodes and 18,892 edges.](image)

Before the trajectory dataset is used for map matching, the noise and redundant GPS points are filtered, and the long trajectories are segmented. The number of GPS points of each trajectory is displayed in Figure 10a, and the sampling rate of each GPS point is displayed in Figure 10b. Most trajectories have fewer than 50 GPS points, and the sampling rates of 63% of GPS points are higher than 30 s.
4.2. Experimental Settings

Ground Truth: A total of 566 high-sampling-rate historical trajectories with sampling intervals less than 30 s are selected to match HMM-based map matching algorithm to obtain the real paths as ground truth. Then, these real paths are used to reconstruct meta paths and build a historical path index library.

Baselines: An HMM-based map matching algorithm [8] and the ST-Matching algorithm [7] are used as the comparison algorithms to verify the effectiveness of HTMM.

Evaluation Criteria: Matching accuracy and matching time are chosen to evaluate HTMM. The matching time is measured by the running time of the program, which does not include the time to build the historical path index library and load the road network data. The matching accuracy is measured by the proportion of the number of correctly matched road segments to the total number of road segments of trajectories, as shown in Equation (14):

$$A_N = \frac{N_c}{N_t}$$  \hspace{1cm} (14)

where $N_c$ represents the number of correctly matched road segments, and $N_t$ represents the number of total road segments.

Test Data: A total of 100 high-sampling-rate trajectories are selected to generate test data, which are first matched to the road network to obtain the real paths as ground truth, and then down-sampled at 60 s, 120 s, 180 s, 240 s, 300 s, 360 s, 420 s, 480 s and 600 s intervals.

4.3. Experimental Results

4.3.1. Comparison of Three Map Matching Algorithms

A change in sampling rate leads to a change of the distance between two adjacent GPS points. Figure 11 shows the average distance between two adjacent GPS points of the test data under different sampling rates. The average distance is 0.67 km when the sampling rate is 60 s but reaches 2.36 km when the sampling rate is 600 s. This introduces problems such as matching interruption and matching error and reduces the matching accuracy.

As shown in Figure 12, the matching accuracy of the three algorithms gradually decreases as the sampling rate decreases. It is clear that HTMM outperforms the HMM-based and ST-Matching algorithms when the sampling rate ranges from 60 s to 300 s. This result may be due to the fact that the paths between adjacent GPS points do not conform to the shortest path principle. This also proves that it is feasible to select local paths according to historical information. However, the accuracy of HTMM drops sharply when the sampling rate ranges from 360 s to 600 s. The main reason for this is that as the path length increases, there are more historical paths to choose from. Drivers also make
choices according to the current situation, not just relying on historical experience. In this case, the current information of GPS points is not fully utilized by the HTMM algorithm.

![Average distance between two adjacent GPS points at different sampling rates.](image)

**Figure 11.** The average distance between two adjacent GPS points at different sampling rates.

![Matching accuracy of three algorithms at different sampling rates.](image)

**Figure 12.** Matching accuracy of three algorithms at different sampling rates.

As shown in Figure 13, with an increase in algorithm complexity, the matching time is increases, so HTMM is inferior to the other two algorithms. Besides the search of global optimal path, the time cost of HTMM is mainly in the calculation of multiple candidate road segments and the retrieval of the historical index library. In particular, the time cost of HTMM increases significantly with a decrease in sampling rate. However, the local candidate path is determined before searching the global matched path. Although the global optimal solution may not be obtained, this operation reduces time cost. It is acceptable to sacrifice time cost for improved matching accuracy.
Figure 13. Matching time of three algorithms at different sampling rates.

4.3.2. Visual Analysis of Typical Situations

Two typical matched paths are selected from the matched results for visual analysis.

(1) Detours

As shown in Figure 14, there are many detours in the real path represented by the green line. After using the ST-Matching algorithm, the detour is replaced by a shorter path when the sampling rate is 60 s. The result become worse when the sampling rate is 120–300 s. This is the result of selecting the shortest path. It is found that the travel time of detours and the shortest paths is very different, so HTMM solves this problem well by calculating the travel time. However, HTMM produces errors in the selection of main roads and auxiliary roads.

(2) Multiple Optimal Paths

Figure 14. Comparison of matched paths of HTMM and ST-Matching: (a) sampling rate of 30 s (real path); (b) sampling rate of 60 s (ST-Matching); (c) sampling rate of 120–300 s (ST-Matching); (d) sampling rate of 120–300 s (HTMM).
As shown in Figure 15, different sampling rate trajectories obtain different matched paths after using the ST-Matching algorithm. This is because the length and direction angle of these optional paths are similar, so the ST-Matching algorithm does not know which one to select. HTMM solves this problem by adding the frequency of the historical paths as the measurement of path selection. It is believed that the result is reliable if the historical path index library is personalized and large enough.

Figure 15. Comparison of matched paths of HTMM and ST-Matching: (a) sampling rate of 30 s (real path); (b) sampling rate of 60–300 s (ST-Matching); (c) sampling rate of 60–120 s (HTMM).

4.3.3. Positioning Error Analysis

In order to verify the effect of HTMM on positioning error correction, the GPS points eliminated by down sampling are taken as the points that need to query the locations. The real locations of these GPS points are known. The meta paths of the matched path are reconstructed to acquire the enter time and the leave time; then, the corrected locations are calculated according to the method introduced in Section 3.4.2. The measurement error of the GPS point is the distance difference between the observation location and its location in the historical real path, the average of which is 15.06 ± 5.57 m. The positioning error is the distance difference between the corrected location and its location in the historical real path, as shown in Table 2. The positioning error is reduced by nearly half when the sampling rate is 60 s.

Table 2. Errors between the corrected GPS points and the real GPS points.

| Sampling Rate | Mean and Std |
|---------------|-------------|
| 60 s          | 7.24 ± 3.81 (m) |
| 120 s         | 7.53 ± 4.95 (m) |
| 180 s         | 8.84 ± 3.42 (m) |
| 240 s         | 10.25 ± 5.28 (m) |
| 300 s         | 13.38 ± 6.35 (m) |

HTMM can eliminate the error perpendicular to roads. However, the error parallel to roads, which is caused by change in speed and other factors, cannot be eliminated.

5. Conclusions

In this paper, a novel map matching algorithm called HTMM is proposed. HTMM is the first map matching algorithm for container positioning and tracking with missing and low-sampling-rate trajectories. Compared with other methods that improve positioning precision at the hardware level, HTMM has a significant cost advantage.

The biggest innovation of HTMM is to propose a local path selection strategy that utilizes travel time and the frequency mined from historical paths. A path reconstruction method is introduced to determine the travel time of historical paths on each road segment before a historical path index library is built. A historical path index library is developed to accelerate the index speed when historical paths are searched in HTMM. Finally, to realize
the query and tracking of containers, a location query and tracking method is presented by road segment retrieval and time–distance interpolation.

After verification on a real trajectory dataset, the matching accuracy of HTMM is proven to be better than that of an HMM-based map matching algorithm and the ST-Matching algorithm, especially when the sampling interval ranges from 30 s to 300 s. The matching accuracy is not improved much when the sampling interval is higher than 300 s. We infer that this is because with changing traffic conditions, the choice of drivers at each intersection between two adjacent GPS points is not necessarily the same every time. The longer the distance between two adjacent GPS points, the greater the diversity. In this case, HTMM only selects one complete historical path with the largest score as the local candidate path.

In the future work, we will further optimize the local candidate path selection strategy. One method is to segment the longer path on the basis of intersections and select a local candidate path for each partial path. Another method is to consider more real-time traffic information to improve the Markov decision process. In addition to this work, we will conduct experiments on a larger trajectory dataset and road network to improve the generalization of HTMM. HTMM will be deployed in online map matching combining with location query and tracking to realize the real-time positioning and tracking of containers, which will greatly improve the positioning precision and safety of container transportation.

Author Contributions: W.L. designed the research roadmap, put forward constructive suggestions and reviewed the manuscript; W.Z. conducted the main research for this paper and wrote the manuscript; C.G. provided suggestions for organizing this manuscript and polished it. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Project (Grant 2019YFB 1600400); and partly by the National Natural Science Foundation of China (Grant 62173263) and Innovation Team Project of Hainan (Grant 621CXTD1013).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://www-users.cse.umn.edu/~tianhe/BIGDATA/UrbanCPS/TruckData/TruckData, accessed on 15 March 2022].

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Choi, H.R.; Moon, Y.S.; Kim, J.J.; Lee, J.K.; Lee, K.B.; Shin, J.J. Development of an IoT-based container tracking system for China’s Belt and Road (B&R) initiative. Marit. Policy Manag. 2018, 45, 388–402.
2. Hao, Y.; Tang, P.; Yang, B. Selection strategy of battery life of reefer container IoT monitoring device. In Proceedings of the 2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI), Shanghai, China, 6–7 September 2018; pp. 1–5.
3. Hsueh, Y.L.; Chen, H.C. Map matching for low-sampling-rate GPS trajectories by exploring real-time moving directions. Inf. Sci. 2018, 433, 55–69. [CrossRef]
4. Nawaz, A.; Huang, Z.; Wang, S.; Akbar, A.; Al-Salman, H.; Gumaei, A. GPS trajectory completion using end-to-end bidirectional convolutional recurrent encoder-decoder architecture with attention mechanism. Sensors 2020, 20, 5143. [CrossRef] [PubMed]
5. Huang, Z.; Qiao, S.; Han, N.; Yuan, C.; Song, X.; Xiao, Y. Survey on vehicle map matching techniques. CAAI Trans. Intell. Technol. 2021, 6, 55–71. [CrossRef]
6. Yu, J.; Yang, Q.; Lu, J.; Han, J.; Peng, H. Advanced map matching algorithms: A survey and trends. Acta Electron. Sin. 2021, 49, 1818–1829.
7. Lou, Y.; Zhang, C.; Zheng, Y.; Xie, X.; Wang, W.; Huang, Y. Map-matching for low-sampling-rate GPS trajectories. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009; pp. 352–361.
8. Newson, P.; Krumm, J. Hidden Markov map matching through noise and sparseness. In Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 4–6 November 2009; pp. 336–343.
9. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, X.; Sun, G. An interactive-voting based map matching algorithm. In Proceedings of the 2010 Eleventh International Conference on Mobile Data Management, Kansas City, MO, USA, 23–26 May 2010; pp. 43–52.
10. Gan, M.; Qing, S.; Liu, X.; Li, D. Review on application of truck trajectory data in highway freight system. *J. Transp. Syst. Eng. Inf. Technol.* **2021**, *21*, 91.
11. Tran-Dang, H.; Kim, D.S. An information framework for internet of things services in physical internet. *IEEE Access* **2018**, *6*, 43967–43977. [CrossRef]
12. Nie, J.; Lyu, J.; Zhou, S.; Huang, R. Intelligent cold chain transportation terminal system based on LEO satellite and Narrowband Internet of Things. *Comput. Syst. Appl.* **2019**, *28*, 119–124.
13. Lu, Y.; Xue, G.; Sun, H.; Xie, Q. Design of multimodal transport information service system based on space-based Internet of Things. *Space Integr. Ground Inf. Netw.* **2020**, *1*, 116–127.
14. Tang, P.; Postolache, O.A.; Hao, Y.; Zhong, M. Reefer container monitoring system. In Proceedings of the 2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, Romania, 28–30 March 2019; pp. 1–6.
15. White, C.E.; Bernstein, D.; Kornhauser, A.L. Some map matching algorithms for personal navigation assistants. *Transp. Res. Part C Emerg. Technol.* **2000**, *8*, 91–108. [CrossRef]
16. Li, X.; Ma, S.; Yang, H.; Zhang, X.N. Vector road matching algorithm based on a topological path. *J. Image Graph.* **2017**, *22*, 0596–0609.
17. Zhu, D.; Liu, Y. *An Incremental Map-Matching Method Based on Road Network Topology*; Geomatics and Information Science of Wuhan University: Wuhan, China, 2017.
18. Li, H.; Kulik, L.; Ramamohanarao, K. Robust inferences of travel paths from GPS trajectories. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 2194–2222. [CrossRef]
19. Yang, C.; Gidofalvi, G. Fast map matching, an algorithm integrating hidden Markov model with precomputation. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 547–570. [CrossRef]
20. Zheng, K.; Zheng, Y.; Xie, X. Reducing uncertainty of low-sampling-rate trajectories. In Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, Arlington, VA, USA, 1–5 April 2012; pp. 1144–1155.
21. Banerjee, P.; Ranu, S.; Raghavan, S. Inferring uncertain trajectories from partial observations. In Proceedings of the 2014 IEEE International Conference on Data Mining, Shenzhen, China, 14–17 December 2014; pp. 30–39.
22. Wu, H.; Mao, J.; Sun, W.; Zheng, B.; Zhang, H.; Chen, Z.; Wang, W. Probabilistic robust route recovery with spatio-temporal dynamics. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1915–1924.
23. Huang, Y.; Rao, W.; Zhang, Z.; Zhao, P.; Yuan, M.; Zeng, J. Frequent pattern-based map-matching on low sampling rate trajectories. In Proceedings of the 2018 19th IEEE International Conference on Mobile Data Management (MDM), Aalborg, Denmark, 25–28 June 2018; pp. 266–273.
24. Bian, W.; Cui, G.; Wang, X. A trajectory collaboration based map matching approach for low-sampling-rate GPS trajectories. *Sensors* **2020**, *20*, 2057. [CrossRef] [PubMed]
25. Gao, X.; Wu, Y.; Guo, L.; Ding, Z.; Chen, J. Personalized map-matching algorithm based on driving preference. *J. Softw.* **2018**, *29*, 3500–3516.