Map-matching algorithm for large-scale low-frequency floating car data

Bi Yu Chen\textsuperscript{a}, Hui Yuan\textsuperscript{ba}, Qingquan Li\textsuperscript{ca}, William H.K. Lam\textsuperscript{d}, Shih-Lung Shaw\textsuperscript{ea} & Ke Yan\textsuperscript{fa}

\textsuperscript{a} State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
\textsuperscript{b} School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
\textsuperscript{c} Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen, China
\textsuperscript{d} Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
\textsuperscript{e} Department of Geography, The University of Tennessee, Knoxville, TN 37996-0925, USA
\textsuperscript{f} International School of Software, Wuhan University, Wuhan 430079, China

Published online: 02 Sep 2013.

To cite this article: Bi Yu Chen, Hui Yuan, Qingquan Li, William H.K. Lam, Shih-Lung Shaw & Ke Yan (2014) Map-matching algorithm for large-scale low-frequency floating car data, International Journal of Geographical Information Science, 28:1, 22-38, DOI: 10.1080/13658816.2013.816427

To link to this article: http://dx.doi.org/10.1080/13658816.2013.816427

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the “Content”) contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims,
Map-matching algorithm for large-scale low-frequency floating car data

Bi Yu Chen\textsuperscript{a,*}, Hui Yuan\textsuperscript{b,a,*}, Qingquan Li\textsuperscript{c,a,*}, William H.K. Lam\textsuperscript{d}, Shih-Lung Shaw\textsuperscript{e,a} and Ke Yan\textsuperscript{f,a}

\textsuperscript{a}State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; \textsuperscript{b}School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; \textsuperscript{c}Shenzhen Key Laboratory of Spatial Smart Sensing and Services, Shenzhen University, Shenzhen, China; \textsuperscript{d}Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong; \textsuperscript{e}Department of Geography, The University of Tennessee, Knoxville, TN 37996-0925, USA; \textsuperscript{f}International School of Software, Wuhan University, Wuhan 430079, China

\textsuperscript{*Corresponding authors. Email: chen.biyu@gmail.com (Bi Yu Chen); yuanhui@whu.edu.cn (Hui Yuan); qqli@whu.edu.cn (Qingquan Li)}

Large-scale global positioning system (GPS) positioning information of floating cars has been recognised as a major data source for many transportation applications. Mapping large-scale low-frequency floating car data (FCD) onto the road network is very challenging for traditional map-matching (MM) algorithms developed for in-vehicle navigation. In this paper, a multi-criteria dynamic programming map-matching (MDP-MM) algorithm is proposed for online matching FCD. In the proposed MDP-MM algorithm, the MDP technique is used to minimise the number of candidate routes maintained at each GPS point, while guaranteeing to determine the best matching route. In addition, several useful techniques are developed to improve running time of the shortest path calculation in the MM process. Case studies based on real FCD demonstrate the accuracy and computational performance of the MDP-MM algorithm. Results indicated that the MDP-MM algorithm is competitive with existing algorithms in both accuracy and computational performance.

Keywords: mobile objects; mobility; map matching

1. Introduction

In recent years, significant attention has been given to the development of intelligent transportation systems (ITS) for alleviating traffic congestion. Among the various subsystems of ITS, traffic surveillance is a fundamental component, with the objectives of monitoring transportation network performance and providing real-time traffic information (Liu \textit{et al.} 2012\textsuperscript{a}). Real-time traffic information enables transportation system operators to adopt dynamic traffic control or management strategies (Sáez \textit{et al.} 2012), and allows travellers to make informed route-choice decisions (Chen \textit{et al.} 2012, 2013\textsuperscript{a, b, c}).

With advances in positioning and wireless communication techniques, floating car data (FCD) has become a major data source for traffic surveillance due to its low operational
cost and large spatial coverage (Li et al. 2011a, Kong et al. 2013, Chang et al., 2013). The floating car system typically makes use of a large fleet of probe vehicles (e.g. taxis) equipped with global positioning system (GPS) equipment. Their locations are collected at regular time intervals to estimate and/or predict the status of the transportation network (Hellinga and Fu 2002, Zou et al. 2012). To reduce the cost of data transmission imposed by large-scale floating car systems, limited location information (i.e. latitude and longitude coordinates) is collected at a low sampling frequency, typically one sample per minute. In the literature, this form of FCD is also sometimes referred to as tracking data or trajectories of moving objects (Güting et al. 2006, Bertolotto et al. 2007).

Apart from its application in traffic surveillance systems, large-scale FCD offers an important opportunity for research into spatial and social behaviour through data mining and knowledge discovery (Lu and Liu 2012). Using detailed spatiotemporal trajectories of probe vehicles, much recently, research effort has been devoted to a better understanding of people’s behaviour, their interactions with each other and with the environment. For example, travellers’ route choice set formulation (Rieser-Schüssler et al. 2012, Ramaekers et al. 2013), urban mobility patterns (González et al. 2008, Liu et al. 2012b) and accessibility patterns (Li et al. 2011b, Chen et al. 2013d), commercial centre attractiveness (Yue et al. 2012) and traffic emission estimation (Chang et al. 2013).

Matching FCD to the road network is a key pre-process of numerous FCD applications. Due to GPS measurement errors and road geometric errors in digital maps, the GPS locations of probe vehicles may not appear on network links. Thus, a map-matching (MM) procedure is required to precisely match this FCD to network links. For convenience, the MM problem for large-scale, low-frequency FCD is hereafter referred to as the ‘FCD-MM problem’.

In the literature, many MM algorithms have been proposed for in-vehicle navigation of single vehicles at high GPS sampling frequency (e.g. 1 sample/second), and differ in their input data and matching techniques (White et al. 2000, Pyo et al. 2001, Greenfeld 2002, Quddus et al. 2006, Chen et al. 2008, Velaga et al. 2009, Yang et al. 2011a). A comprehensive review of such MM algorithms appears in Skog and Handel (2009).

Traditional MM algorithms for in-vehicle navigation cannot be directly used to solve the FCD-MM problem for low sampling frequencies. For example, if the speed of a probe vehicle is 36 km/h, the sampling interval is 2 minutes and the distance between two consecutive GPS points is 1.2 km, the vehicle may have passed through several network links in that time. How to identify the actual route taken by the vehicle poses a challenge for traditional MM algorithms. In addition, in the floating car system, generally only the latitude and longitude coordinates are recorded. Matching these FCD to the limited positioning information is extremely difficult for many existing MM algorithms using rich positioning information, such as heading, vehicle speed and additional information from dead reckoning and inertial navigation devices. Moreover, the real-time matching of mass FCD poses another challenge for existing MM algorithms, since the number of probe vehicles may be very large (e.g. more than 10,000 taxis in Wuhan, China).

Considering the increasing importance of FCD applications, it was, therefore, necessary to develop new FCD-MM algorithms for solving these problems. The present study paper has precisely addressed the FCD-MM problem by devising an efficient and effective FCD-MM algorithm. The remainder of this paper is organised as follows. In the next section, the unique features of our proposed algorithm are introduced after a careful review of the existing FCD-MM algorithms. The FCD-MM problem is defined in Section 3. The proposed FCD-MM algorithm is described in Section 4. Case studies using real FCD are reported in Section 5. Finally, the conclusions, together with future research recommendations, are given in Section 6.
2. Literature review

Marchal et al. (2005) presented the first FCD-MM algorithm for an offline matching GPS log, but with a high sampling frequency. In their algorithm, only latitude and longitude coordinates are used to match FCD by adopting a 'curve-to-curve' method. To improve MM accuracy in dense areas, the algorithm adopts the multiple hypothesis technique (MHT) (Pyo et al. 2001) to keep track of all the possible candidate routes during the MM process. Although the algorithm obtains accurate MM results, it is computationally intensive since the number of candidate routes grows exponentially with the MM process. To make it computationally tractable, a heuristic approach is adopted by tracking only $K$ candidate routes with minimum spatial proximity scores. However, this heuristic approach may miss the best matching route completely when $K$ is small. For choosing an appropriate value of $K$, it is, therefore, necessary to review the trade-off between accuracy and computational performance.

Following the work of Marchal et al. (2005), Li and Huang (2010) proposed an FCD-MM algorithm for large-scale, low-frequency FCD. With low sampling frequency FCD, a vehicle may pass several links during the sampling interval. Thus, the shortest path algorithm was employed to generate candidate routes between two consecutive GPS points, but unlike the spatial proximity approach in Marchal et al. (2005), Li and Huang (2010) adopted a shape similarity score to measure the degree of geometric similarity between the vehicle trajectory and candidate routes. However, the heuristic approach of MHT was employed to deal with the large-candidate route set problem.

Wang et al. (2011) also proposed an FCD-MM algorithm using the framework of Marchal et al. (2005) in which a fuzzy logic approach was used to generate a synthesised route evaluation score based on multiple decision factors, including spatial proximity, vehicle speed and heading. However, the speed and heading information may not be available for all floating car systems. To prevent the exponential growth of the number of candidate routes in the MHT, the vehicle trajectories are divided and processed into 5 min intervals. Nevertheless, the separation of vehicle trajectories may lead to the disconnection of routes generated over different time intervals.

The FCD-MM algorithm of Lou et al. (2009) addressed the large-candidate route set issue of MHT. Rather than incrementally generating candidate routes during the MM process, the algorithm constructs a candidate graph to determine the ‘global’ best route for the whole vehicle trajectory. In the proposed candidate graph, each graph node represents a candidate point for a GPS point, and each graph edge represents the shortest path between two consecutive candidate points. The global best matching route is determined by finding the longest path with the best route evaluation score in the candidate graph. Although this approach avoids generating a set of candidate routes, the algorithm can only be used for offline FCD matching after the whole candidate graph has been constructed.

As an alternative to MHT, Miwa et al. (2012) proposed an FCD-MM algorithm in which a GPS point is matched to the network link by considering its previous matched point and the next GPS point. As no candidate route set is constructed during the MM process, the algorithm has good computational performance. However, the results are affected if previous GPS points are incorrectly matched in dense urban areas. Newson and Krumm (2009) proposed a map-matching algorithm based on the hidden Markov model. Nevertheless, the MM accuracy and computational performance of this algorithm have not been investigated in comparison to other FCD-MM algorithms.

Along the line of Marchal et al. (2005), the present study proposes a multi-criteria dynamic programming map matching (MDP-MM) algorithm for online matching of
large-scale low-frequency FCD. The proposed MDP-MM algorithm extends previous studies in the following respects:

(1) The large-candidate route set issue of MHT is addressed by a multi-criteria dynamic programming technique, utilising route evaluation score and network topology as the two criteria for determining candidate routes at each GPS point. By this technique, only the candidate route with the minimum route evaluation score is required to be stored at each candidate location of the GPS point. Consequently, the size of each candidate route is reduced to the number of candidate locations within the error region of the GPS point (instead of increasing exponentially with the MM process), yet guarantees that the global best matching route is obtained.

(2) A modified shortest path procedure is developed in the proposed MDP-MM algorithm for efficiently finding the candidate route at each candidate location. In the FCD-MM problem, the shortest path procedure is continually repeated to determine the candidate route between two consecutive GPS points. Given the huge number of GPS points in the FCD-MM problem, the calculation of the shortest path is computationally intensive. In this study, several useful techniques are introduced to improve running times for the shortest path calculation, including shortest path finding without label initialisation, and determination of the best candidate route from multiple origins to the same destination.

(3) Case studies using real FCD are carried out to examine the accuracy and computational performance of the proposed MDP-MM algorithm. The four existing FCD-MM algorithms (Marchal et al. 2005, Newson and Krumm 2009, Li and Huang 2010, Miwa et al. 2012) are also implemented for comparison purposes. Results of the case studies indicated that the proposed MDP-MM algorithm is competitive with existing FCD-MM algorithms with respect to both MM accuracy and computational performance, and also demonstrate that the proposed MDP-MM algorithm is capable of matching large-scale, low-frequency FCD in real time.

3. Problem statement

As discussed above, FCD-MM algorithms determine the vehicle trajectory in road networks using GPS data and digital road network data. In the present study, a road network is represented as a directed graph $G = (N, A, \Psi)$, where $N$ is the set of nodes, $A$ is the set of links and $\Psi$ is the set of allowed movements (Yang et al., 2011b, 2013). Any location $l^i$ in the road network can be represented as $l^i = \{a^i, \theta^i\}$ using the linear reference system technique, where $\theta^i \in [0, 1]$ indicates a relative position on a link $a^i \in A$. For example, $\theta^i = 0$, $\theta^i = 0.5$ and $\theta^i = 1$ represent the beginning, middle and end of a link $a^i$.

A GPS record measures the location of a vehicle at a given moment in time. It has the following basic attributes: latitude, longitude, altitude, timestamp, heading, instantaneous speed, the number of satellites in view, etc. In the floating car system, generally only the location (i.e. latitude and longitude coordinates) and timestamp are recorded in order to reduce the data transmission cost. The location and timestamp of a GPS record $i$ are denoted by $p^i$ and $t^i$, respectively. Due to satellite signal blockage and multipath effects, the positioning accuracy of $p^i$ in an urban area is in the range 0–40 m at the 95% confidence level (Quddus et al. 2007). In this case, the vehicle is likely to be travelling on several candidate links within the error region of $p^i$ (see Figure 1). The candidate locations on these links are denoted by $L^i = \{l^i_1, ..., l^i_n\}$. 
Consider a sequence of GPS records for a vehicle denoted by \( \{(p^1, t^1), \ldots, (p^i, t^i)\} \). As the sampling frequency in a floating car system is generally low, the vehicle may pass several links during the time period. The aim of the FCD-MM algorithm is to determine the best matching route \( r^*_i \) (see Figure 1), including not only a sequence of corresponding network locations \( l^1_n, \ldots, l^i_n \), but also a sequence of intermediate network nodes.

4. Multi-criteria dynamic programming map-matching algorithm

4.1. Multi-criteria dynamic programming technique

As discussed above, the MHT is adopted in FCD-MM algorithms to improve the map-matching accuracy in dense urban areas. Using this technique, multiple candidate routes are maintained at each GPS point. Figure 1 illustrates a simple example of this technique: three GPS points \( (p^1, p^2, p^3) \) are recorded for a probe vehicle, and each GPS point has two candidate locations within its error region. Using MHT, all possible candidate routes are maintained at each GPS point during the MM process. After the termination of the MM process, the best matching route \( r^*_i \) with minimum route evaluation score \( f(r^*_i) \) is determined from the generated candidate routes.

Spatial proximity was recognized as an effective route evaluation criterion that measures the spatial closeness of a vehicle trajectory against a candidate route (Marchal et al. 2005). As shown in Figure 1, the Euclidean distance between GPS \( p^j \) and candidate location \( l^j_n \) is denoted by \( h^j_n \). Let \( r^j_{n,1} \in R^j \) be a candidate route passing through candidate location \( l^j_n \). Then, its spatial proximity index, denoted by \( h(r^j_{n,1}) \), can be expressed by the sum of \( h^j_n \) values along the candidate route, given by:

\[
h(r^j_{n,1}) = \sum_{j \neq i} h^j_n
\]  

For instance, the spatial proximity score for the best matching route in Figure 1 is \( h(r^3_{1,1}) = h^1_1 + h^2_1 + h^3_1 \).

Although this technique is simple and effective, the size of candidate routes grows exponentially in the MM process (e.g. illustrated by the four candidate routes at \( p^2 \) and eight candidate routes at \( p^3 \) in Figure 1). The exponential growth of candidate routes makes the MHT technique computationally intractable when attempting to use it to solve a large-scale FCD-MM problem.

To prevent the exponential growth of candidate routes in this way, a multi-criteria dynamic programming technique is proposed to eliminate dominated routes. Let \( r^j_1 \in R^j \) be a candidate route maintained at GPS point \( p^j \). For a subsequent GPS point \( p^i \),
let $r_1' = r_1' \oplus r_2'$ be a candidate route passing along route $r_1'$. The dominated route is defined as follows.

**Definition 1.** A route $r_1' \in R^i$ dominates another route $r_2' \in R^i$, if and only if its route evaluation score satisfies $f(r_1' = r_1' \oplus r_2') < f(r_2' = r_2' \oplus r_2')$ for any subsequent GPS point $p'$.

Based on Definition 1, the dominated condition is proved below, using network topology and spatial proximity score as two independent criteria. Let $r_{n,1} \in R^i$ and $r_{n,2} \in R^i$ be two candidate routes passing through the same candidate location $l_n$ of GPS point $p'$.

**Proposition 1.** A route $r_{n,1} \in R^i$ dominates another route $r_{n,2} \in R^i$, if $h(r_{n,1}) < h(r_{n,2})$.

**Proof.** Given a subsequent GPS point $p'$, suppose that there exists a route $r_m'$ connecting candidate locations $l_n'$ and $l_m'$. We have $h(r_{n,1} = r_{n,1}' \oplus r_m') = h(r_{n,1}) + h(r_m' - h_n') < h(r_{n,2}) + h(r_m') - h_n' = h(r_{n,2} = r_{n,2}' \oplus r_m')$ for any candidate location of any subsequent GPS point $p'$. According to Definition 1, we have $r_{n,1}'$ dominates $r_{n,2}'$.

Apart from spatial proximity, route evaluation criteria with other properties can also be used, including a linear combination of the spatial proximity $h(r_{n,1})$ and travel distance $d(r_{n,1})$, as:

$$g(r_{n,1}) = \eta h(r_{n,1}) + d(r_{n,1})$$

where $\eta \geq 0$ is a weighting parameter. The dominated condition can readily be proved as follows, taking $g(r_{n,1})$ as the route evaluation score.

**Proposition 2.** A route $r_{n,1} \in R^i$ dominates another route $r_{n,2} \in R^i$, if $g(r_{n,1}) < g(r_{n,2})$.

**Proof.** Given a subsequent GPS point $p'$, suppose that there exists a route $r_m'$ connecting candidate locations $l_n'$ and $l_m'$. We have $g(r_{n,1} = r_{n,1}' \oplus r_m') = \eta h(r_{n,1}) + h(r_m') - h_n') + d(r_{n,1}) + d(r_m')$, and thus $g(r_{n,1}) = r_{n,1}' \oplus r_m') < \eta h(r_{n,2}) + h(r_m') - h_n') + d(r_{n,2}) + d(r_m') = g(r_{n,2} = r_{n,2}' \oplus r_m')$ for any candidate location of any subsequent GPS point $p'$. According to Definition 1, we have $r_{n,1}'$ dominates $r_{n,2}'$.

With Proposition 2, the FCD-MM problem can be solved by the multi-criteria dynamic programming technique in terms of route evaluation score and network topology. At each candidate location $l_n'$ of a GPS point $p'$, only one candidate route $r_n'$ with minimum evaluation score $g(r_n')$ is maintained as a non-dominated route. Overall, $|L'|$ non-dominated routes are retained at the GPS point $p'$, where $|L'|$ is the number of candidate locations within the error region of GPS point $p'$. According to Definition 1, other dominated routes can be eliminated without further consideration in the MM process, since they cannot be a part of the best matching route.

### 4.2. Solution algorithm

In this study, a multi-criteria dynamic programming map-matching (MDP-MM) algorithm is proposed for matching large-scale, low-frequency FCD. The framework of the proposed MDP-MM algorithm in Figure 2 consists of four steps: initialisation, trajectory tracking, candidate location determination and candidate route generation.
In Step 1 (initialisation), the network data is pre-loaded into central memory in order to accelerate the MM process. A 2D R-tree spatial index (Guttman, 1984) is constructed for all network links using their geometric shape, in order to facilitate the spatial query in the road network. An adjacent list data structure is also constructed in this step to load the network topology into central memory in order to find the shortest path in the road network.

In Step 2 (trajectory tracking), real-time GPS records from the floating car system are handled vehicle by vehicle.

In Step 3 (candidate location determination), candidate locations for the vehicle are determined for each GPS record \((p_i, t_i)\). First, a set of candidate links (denoted by \(A^i = \{a^i_{qw}, \ldots\}\)) within the error region of the current GPS point \(p_i\) is retrieved by a spatial query. Then, the set of candidate locations \(L^i = \{l^i_1, \ldots, l^i_n\}\) is determined by projecting \(p_i\) onto the retrieved candidate links and minimising the Euclidean distance between \(p_i\) and \(l^i_n\) (i.e., the \(h^i_n\) value shown in Figure 1).

In Step 4 (candidate route generation), candidate routes stored from the previous GPS point \(p_{i-1}\) (i.e. \(R^{1,i-1} = \{r_{1,i-1}^{1,i-1}, \ldots, r_{s,i-1}^{1,i-1}\}\)) are extended to generate candidate routes at the current GPS point \(p_i\) (i.e. \(R^{1,i} = \{r_{s,i}^{1,i}, \ldots, r_{s,n}^{1,i}\}\)). Using the multi-criteria dynamic programming technique described above, \(|L^i|\) non-dominated routes are generated as candidate routes at the current GPS point \(p_i\). After the candidate route \(r_{s,n}^{1,i}\) is determined, the average speed from \(l^i_{n-1}\) and \(l^i_n\) is calculated and validated by the speed limit \(v_{\text{max}}\) of the road network.

Steps 2–4 are run continuously until all GPS records are matched to the road network. The steps of the MDP-MM algorithm are described in detail below.

**MDP-MM algorithm**

**Step 1. Initialisation:**

Construct an R-tree spatial index for the road network.

Load network topology into main memory by constructing an adjacent list.
Step 2. Trajectory tracking:
If all GPS records have been matched, then Stop.
Retrieve a GPS record \((p', r')\) of a probe vehicle.

Step 3. Candidate location determination:
Search candidate links \(A' = \{a'_1, \ldots, a'_n\}\) within the error region of \(p'\).
Determine candidate locations \(L' = \{l'_1, \ldots, l'_n\}\) by projecting \(p'\) onto candidate links \(A'\).

Step 4. Candidate route generation:
For each candidate location \(l'_n \in L'\)
Call procedure \(r'_n := \text{FCDSP}(R^{i-1}, l'_n)\) to determine the best candidate route \(r'_n\)
with minimum \(g(r'_n)\) value to current candidate location \(l'_n\).
If travel speed violates speed limit \(d(r'_{n-1})/(t'_{n-1}) > v_{\text{max}}\), then \(r'_n := \text{null}\).
End for
Go to Step 2.

In the proposed MDP-MM algorithm, the FCD shortest path (FCDSP) procedure is to find the candidate route \(r'_n\) with minimum \(g(r'_n)\) value to current candidate location \(l'_n\). In the study, it is assumed that the probe vehicle always takes the shortest path between two consecutive GPS points. This assumption seems reasonable, because the time interval of two consecutive GPS records is relatively short (Wang et al. 2011, Miwa et al. 2012). Based on this assumption, the link-based Dijkstra’s algorithm (Gutierrez and Medaglia 2008), which explicitly considers the turn restrictions in the road network, can be employed to determine the route between two candidate locations. In the proposed FCDSP procedure, Dijkstra’s algorithm is modified in several aspects as follows.

First, in the proposed FCDSP procedure, the candidate route \(r'_n\) is determined by a single path finding process. Intuitively, to obtain \(r'_n, |L^{i-1}|\) shortest paths from all previous candidate locations \(\forall l_{n-1} \in L^{i-1}\) to the current candidate location \(l'_n\) should be calculated by repeatedly using Dijkstra’s algorithm. Then, \(r'_n\) can be obtained by choosing the route with minimum \(g(r'_n)\) value of all calculated shortest paths. In this study, Dijkstra’s algorithm was modified to obtain \(r'_n\) in a single path finding process. In Step 2, all routes \(\forall r_{k-1}' \in R^{i-1}\) at previous candidate locations are added into the priority queue (denoted by \(SE\)) for path extension. The path cost of \(r_{k-1}'\) is set as \(g(r_{k-1}') + \eta h_r'). With the path extension process in Step 4, the route \(r'_n\) with minimum \(g(r'_n)\) value can be determined when the candidate location \(l'_n\) is reached in Step 3.

Second, the FCDSP procedure accelerates the shortest path finding process by avoiding the label initialisation step that is required. In Dijkstra’s algorithm label, the initialisation step is required to set null labels to all network links before carrying out the shortest path finding process. As the sampling time interval of FCD is relatively short, the probe vehicle passes only a few links between \(l_{k-1}'\) and \(l'_n\). Dijkstra’s algorithm only explores a very small number of links and, therefore, it is not necessary to visit all network links in every path finding process. In the FCDSP procedure, the ‘label initialisation’ step is avoided by assigning a unique ID (denoted by \(SPID\)) to each path finding process. A process ID attribute, denoted by \(UID(a_{sp})\), is also assigned to each network link \(a_{sp}\) to indicate which path finding process its label belongs to. If \(UID(a_{sp})\) is equal to \(SPID\), the label associated with this link is generated by the current path finding process; otherwise by the previous path finding process. In this way, the label initialisation step can be avoided.

Finally, in the proposed FCDSP procedure, the origin and destination (O-D) of the shortest path are found not only at network nodes, but also on network links. In Dijkstra’s
algorithm, O-D are located only at nodes; however, candidate locations of a GPS point can then be located anywhere on a network link. The proposed FCDSP procedure addresses this issue by adding dummy nodes and links into the road network (i.e. adjacent list data structure): see Steps 1 and 5 below.

**FCDSP procedure**

**Inputs:** Previous candidate routes \( R^{\perp-1} = \{r^{\perp-1}_1, \ldots, r^{\perp-1}_\lambda \} \), current candidate location \( l^\perp_n = \{a_{wq}, \theta^\perp_n\} \) and unique process ID \( \text{SPID} \).

**Returns:** Candidate route \( r^0_n \) at candidate location \( l^\perp_n \).

**Step 1.** Add dummy nodes and links:

Set the set of temporary nodes \( \tilde{N} := \phi \) and the set of temporary links \( \tilde{A} := \phi \).

For each \( \ell^\perp_k \in L^\perp-1 \) (\( \ell^\perp_k = \{a_{uk}, \theta^\perp_k\} \))

- If \( a_{wq} = a_{uk} \) then
  - Create a link \( a_{\tilde{k}k} \) from \( \ell^\perp_k \) to \( l^\perp_k \) and set link distance \( d_{\tilde{k}k} := (\theta^\perp_k - \theta^\perp_k)d_{uk} \).
  - Else
  - Create a link \( a_{\tilde{k}k} \) from \( \ell^\perp_k \) to head node \( n_k \) and set \( d_{\tilde{k}k} := (1 - \theta^\perp_k)d_{uk} \).

  End if

Set \( \tilde{N} := \tilde{N} \cup \{l^\perp_n\} \) and \( \tilde{A} := \tilde{A} \cup \{a_{\tilde{k}k}\} \).

**End for**

Create a link \( a_{wt} \) from tail node \( n_w \) to \( l^\perp_n \) and set link distance \( d_{wt} := \theta^\perp_n d_{wq} \).

Set \( \tilde{N} := \tilde{N} \cup \{l^\perp_n\} \) and \( \tilde{A} := \tilde{A} \cup \{a_{wt}\} \).

Insert \( \tilde{N} \) and \( \tilde{A} \) into network \( G \).

**Step 2.** Initialisation:

- Create a priority queue \( SE := \phi \).
  - For each \( r^{\perp-1}_\lambda \in R^{\perp-1} \)
    - Create a route \( \tilde{r}^{\perp}_n := r^{\perp-1}_\lambda \) and set \( g(\tilde{r}^{\perp}_n) := g(r^{\perp-1}_\lambda) + \eta^* h^*_n \).
    - Set \( SE := SE \cup \{\tilde{r}^{\perp}_n\} \).

**End for**

**Step 3.** Path selection:

- If \( SE = \phi \), then Go to Step 5; otherwise, continue.
  - Select \( r^*_n \) with minimum \( g(\tilde{r}^*_n) \) value from \( SE \), and set \( SE := SE \setminus \{r^*_n\} \).
  - If \( n_k = l^\perp_n \), then set \( r^*_n := r^*_n \) and \( g(r^*_n) := g(r^*_n) \) and Go to Step 5; otherwise continue.
    (\( n_k \) denotes the last node of \( r^*_n \))

**Step 4.** Path extension:

- For each movement \( \psi_{akv} = \{a_{uk}, a_{kv}\} \) (\( a_{uk} \) denotes the last link of \( r^*_n \))
  - Create a route \( r^*_{n,1} := r^*_n \oplus a_{uv} \) and set \( g(r^*_{n,1}) := g(r^*_n) + d(a_{uv}) \).
  - If \( r^*_{n,1} = \phi \) or \( \text{UID}(a_{kv}) \neq \text{SPID} \), set \( r^*_{n,1} := r^*_{n,1} \), \( g(r^*_{n,1}) := g(r^*_n) \), \( SE := SE \cup \{r^*_{n,1}\} \) and \( \text{UID}(a_{kv}) := \text{SPID} \).
  - Else if \( g(r^*_{n,1}) < g(r^*_n) \), set \( r^*_{n,1} := r^*_{n,1} \) and \( g(r^*_{n,1}) := g(r^*_{n,1}) \), and update order of \( r^*_{n,1} \) in \( SE \).

**End if**

**End for**

Go to Step 3.

**Step 5.** Remove dummy nodes and links:

- Remove \( \tilde{N} \) and \( \tilde{A} \) from network \( G \), and Stop.
5. Case study

The accuracy and computational performance of the proposed MDP-MM algorithm were investigated for real-world case studies. The MDP-MM algorithm was coded in the Visual C# programming language. Fibonacci heap (F-heap) data structure (Fredman and Tarjan 1987) was implemented as the priority queue in the FCDSP procedure. The radius of the error region of each GPS point was set as 40 m.

To compare and benchmark the proposed algorithm, the $K$ spatial proximity ($KSP-MM$) algorithm (Marchal et al. 2005), the $K$ shape similarity ($KSS-MM$) algorithm (Li and Huang 2010), moving window ($MW-MM$) algorithm (Miwa et al. 2012) and hidden Markov map matching ($HMM-MM$) (Newson and Krumm 2009) were also implemented in Visual C#. It should be noted that the original $KSP-MM$ algorithm was designed for matching GPS logs with a high sampling frequency. To handle low-frequency FCD, a shortest path finding procedure was added in the original $KSP-MM$ algorithm. The $K$ parameter was set at 10 in both $KSP-MM$ and $KSS-MM$ algorithms. For the $MW-MM$ algorithm, a ‘link cost and route evaluation’ approach (Miwa et al. 2012) was adopted. The link-based Dijkstra’s algorithm (Gutierrez and Medaglia 2008) was adopted as the shortest path finding procedure in the three FCD-MM algorithms. All experiments were conducted on a desktop computer with an Intel dual-core 3.1 GHz CPU (only a single processor used), 8 GB RAM, running on Windows 7 operating system.

5.1. Map matching accuracy

To validate the MM accuracy of the proposed MDP-MM algorithm, the MM results for 13,700 km of 2 s GPS data were collected from a logistics company for fleet management in Wuhan, China. As shown in Figure 3, the Wuhan road network consists of 19,306 nodes and 46,757 links.

The 2 s GPS data were re-sampled into time intervals from 10 s to 120 s. The FCD-MM algorithms were utilised to match the original and re-sampled GPS data to the Wuhan road network. The MM results for the original data were regarded as the ground true values for validating the accuracy of the re-sampled data. In this study, the accuracy ratio of points matched (ARP), a commonly used index in the literature (Miwa et al. 2012), was employed to quantify the MM accuracy, given by:

$$ARP = \frac{N_{correct}}{N_{original}} \times 100\% \quad (3)$$

where $N_{correct}$ is the number of correctly matched points and $N_{original}$ is the number of points in the original 2 s GPS data.

The MM accuracy of the proposed MDP-MM algorithm was investigated for various $\eta$ values. The $\eta$ parameter in the route evaluation function, Equation (2), represents the weighting between spatial proximity and travel distance. Table 1 reports the MM accuracy when sampling time interval $\Delta = 40$ s. As Table 1 shows, the combination of spatial proximity and travel distance slightly improved the MM accuracy of the MDP-MM algorithm. When only travel distance was used (i.e. $\eta = 0$), the ARP value was 90.53%. Large values of $\eta$ (e.g. $\eta = 10$) approximate to the scenario in which only spatial proximity is used in the route evaluation function: in this scenario, the ARP value was 91.52%. By combining travel distance and spatial proximity, the ARP value was slightly improved to 91.57% when $\eta = 0.25$. 

Table 1. The accuracy of MDP-MM algorithm under various $\eta$ values.

<table>
<thead>
<tr>
<th>$\eta$ value</th>
<th>0</th>
<th>0.25</th>
<th>0.5</th>
<th>1</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARP value (%)</td>
<td>90.53</td>
<td>91.57</td>
<td>91.55</td>
<td>91.52</td>
<td>91.52</td>
</tr>
</tbody>
</table>

Figure 4 illustrates the MM accuracy of the proposed MDP-MM algorithm and four existing FCD-MM algorithms. The $\eta$ parameter was set as 0.25 for the MDP-MM algorithm. Figure 4 shows that the accuracy of all five FCD-MM algorithms decreased with increasing sampling time interval $\Delta$. For example, the ARP value was 96.76% when $\Delta = 10$ s using the MDP-MM algorithm, 91.57% when $\Delta = 40$ s, 89.29% when $\Delta = 60$ s and 81.87% when $\Delta = 120$ s.

This result was expected since, in this study, interpolation of the vehicle trajectory between two consecutive GPS points was based on the assumption that the probe vehicle was travelling along the shortest path. This assumption is reasonable when $\Delta$ is small, as the distance between two consecutive GPS points is short and there is only a small chance that the vehicle would have used an alternative route, but with larger $\Delta$, the greater distance made the choice of more alternative routes possible (see Figure 5).

It is seen in Figure 4 that the proposed MDP-MM algorithm obtained better MM results than either KSP-MM or KSS-MM. For example, when sampling time interval $\Delta = 40$ s, the ARP value of the MDP-MM algorithm was 91.57%, which is 4.06% and 19.21% higher than for the KSP-MM and KSS-MM algorithms, respectively. This result is mainly due to their different means of keeping candidate routes. To address the large candidate route...
issue of MHT, the \textit{KSP-MM} algorithm adopts the simple heuristic approach of keeping $K$ candidate routes at each GPS point. This heuristic approach, however, cannot be guaranteed to produce the best matching route. The multi-criteria dynamic programming technique employed in the proposed \textit{MDP-MM} algorithm maintains the candidate route with the minimum route evaluation score at each candidate location. This technique guarantees that the best matching route is determined and, therefore, it has a better MM accuracy than the \textit{KSP-MM} algorithm. Figure 4 shows that the \textit{KSP-MM} algorithm obtained much better
MM results than KSS-MM, despite their use of the same heuristic approach. This indicates that the spatial proximity approach used in the KSP-MM algorithm is more effective than the spatial similarity approach in the KSS-MM algorithm.

Figure 4 shows that the proposed MDP-MM algorithm outperformed the MW-MM algorithm: for instance, the ARP value of the MDP-MM algorithm was 6.04% higher than that for the MW-MM algorithm when \( \Delta = 40 \) s. An illustration of this is given in Figure 6, which shows that the MW-MM algorithm matched a GPS point only when based on its previous matched point and the next GPS point. In a dense road network, the incorrect match of the previous GPS point (e.g. the third point in Figure 6) significantly affected the accuracy of matching the current GPS point (e.g. the fourth point in Figure 6). The proposed MDP-MM algorithm keeps track of all possible candidate routes during the MM process in order to match GPS points based on the whole trajectory of the vehicle. In this way, the proposed algorithm is robust to a mismatch of intermediate GPS points.

Figure 4 also reports the MM accuracy of the HMM-MM algorithm. Two adjustable parameters in the HMM-MM algorithm, \( \sigma_z \) and \( \beta \), were calibrated and set as 15 and 25, respectively, for the testing data. It can be seen from Figure 4 that the HMM-MM algorithm outperforms the MW-MM algorithm. For instance, the ARP value of the HMM-MM algorithm was 5.26% higher than that of the MW-MM algorithm when sampling time interval \( \Delta = 40 \) s. It should be noted that the MDP-MM algorithm was still the best performer for all sampling time intervals.

### 5.2. Computational performance

The computational performance of the proposed MDP-MM algorithm was investigated using real FCD from Wuhan, China. The floating car system in Wuhan utilises 11,248 taxis...
Table 2. Computational performance of MDP-MM algorithm.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Without I/O</th>
<th>With I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Running time (seconds)</td>
<td>Points/second</td>
</tr>
<tr>
<td>MDP-MM</td>
<td>30.84</td>
<td>5582.17</td>
</tr>
<tr>
<td>Algorithm 1</td>
<td>356.18</td>
<td>483.33</td>
</tr>
<tr>
<td>Algorithm 2</td>
<td>1368.97</td>
<td>125.75</td>
</tr>
</tbody>
</table>


As probe vehicles and its sampling time interval is about 40 s. The FCD is collected in real time for estimating (and/or predicting) travel time information in the Wuhan road network. In this case study, 15-minute FCD on 28 December 2008 (Sunday) was collected during 8:30–8:45 am during the morning peak period. Totally, 172,154 GPS points from 10,245 taxis were collected and stored on an SQL Server 2005 database along with matched vehicle trajectories.

Table 2 gives the computational performance of the MDP-MM algorithm. It shows that the MDP-MM algorithm matched FCD inside the 15-minute data collection period \((754.66/60 = 12.6\) min) when database I/O was considered \((754.66 – 30.84 = 723.82 \text{ s})\). For some online FCD applications without I/O in the database, the computational time of the MDP-MM algorithm was reduced to 30.84 s \((5582.17 \text{ points/s})\). This indicates that the MDP-MM algorithm is capable of matching large-scale FCD in real time.

The computational advantages of the FCDSP procedure were demonstrated by developing and running two modified versions of the MDP-MM algorithm (Table 2). Algorithm 1 modified the FCDSP procedure by including a label initialisation step. Algorithm 2 used the classical link-based Dijkstra’s algorithm, including the label initialisation step. Table 2 shows that the MDP-MM algorithm ran 11.55 (356.18/30.84) times faster than Algorithm 1; this was because the MDP-MM algorithm avoided unnecessary label initialisation in its path finding procedure. It is also seen in Table 2 that Algorithm 1 ran 3.84 (1368.97/356.18) times faster than Algorithm 2. This result was expected: if we suppose that \(|L_i^{-1}|\) and \(|L_i|\) candidate routes were maintained at GPS points \(p_{i-1}\) and \(p_i\), respectively, then, to determine \(|L_i'|\) routes at \(p'_i\), Algorithm 2 had to run Dijkstra’s algorithm \(|L_i^{-1}| \times |L_i'|\) times, while Algorithm 1 only ran the FCDSP procedure \(|L_i'|\) times. This demonstrates that the MDP-MM algorithm gives significant improvement in MM efficiency by reducing the number of shortest path findings and avoiding unnecessary label initialisation.

Table 3 shows the computational performance of the four existing FCD-MM algorithms MW-MM, KSS-MM, KSP-MM and HMM-MM (note that, since I/O cost of about 730 s was relatively fixed, it is not included in the table). The table shows that the computational performance of each of the three algorithms was also significantly improved by omitting the label initialisation step in the path finding procedure: the running time of KSP-MM, for example, was reduced by 28.35 times from 4271.22 s to 150.64 s.

Table 3 also demonstrates that the proposed MDP-MM algorithm was superior to the four existing FCD-MM algorithms. For example, the computational time required by the MDP-MM algorithm was 20.47% (30.84/150.64) of the time taken by the KSP-MM algorithm. This is because MDP-MM algorithm minimises the number of candidate routes maintained at each GPS point \(p'_i\). In the KSP-MM algorithm, \(K = 10\) candidate routes have
Table 3. Computational performance of FCD-MM algorithms without I/O cost.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Running time (seconds)</th>
<th>Points/second</th>
<th>Running time (seconds)</th>
<th>Points/second</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDP-MM</td>
<td>30.84</td>
<td>5582.17</td>
<td>356.18</td>
<td>483.33</td>
</tr>
<tr>
<td>HMM-MM</td>
<td>62.31</td>
<td>3133.30</td>
<td>1414.65</td>
<td>138.01</td>
</tr>
<tr>
<td>MW-MM</td>
<td>73.43</td>
<td>2344.62</td>
<td>798.73</td>
<td>215.53</td>
</tr>
<tr>
<td>KSS-MM</td>
<td>145.89</td>
<td>1180.07</td>
<td>4161.07</td>
<td>41.37</td>
</tr>
<tr>
<td>KSP-MM</td>
<td>150.64</td>
<td>1142.82</td>
<td>4271.22</td>
<td>40.31</td>
</tr>
</tbody>
</table>

to be maintained at each GPS point $p_i$. Using the multi-criteria dynamic programming technique, the $MDP-MM$ algorithm is only required to keep $|L_i|$ candidate routes at $p_i$, and generally $|L_i| < 10$.

6. Conclusions

This investigation of the FCD-MM problem aimed at matching large-scale low-frequency floating car data (FCD) to the road network. To solve the FCD-MM problem, a multi-criteria dynamic programming map matching ($MDP-MM$) algorithm is proposed, based on the multiple hypothesis technique (MHT). The large candidate route issue of MHT was addressed by the multi-criteria dynamic programming technique in which route evaluation score and network topology are two criteria for determining the candidate routes at each GPS point. Using this technique, the number of candidate routes at each GPS point is reduced to the number of candidate locations within the error region of the GPS point, and the best matching route is guaranteed.

Two useful techniques were developed to improve the path finding procedure. The first is to avoid unnecessary label initialisation in the path finding procedure. The second technique is to determine the candidate route at a candidate location by a single path-finding process, instead of repeatedly finding shortest paths from previous candidate locations, as done in the classical Dijkstra’s algorithm.

To demonstrate the applicability of the proposed $MDP-MM$ algorithm, two case studies using real FCD in Wuhan, China were carried out. The results indicated that the proposed $MDP-MM$ algorithm was superior to existing FCD-MM algorithms in both MM accuracy and computational performance (Marchal et al. 2005, Newson and Krumm 2009, Li and Huang 2010, Miwa et al. 2012). The results of the case studies also suggested that the trajectories of probe vehicles were well constructed (i.e. the accuracy is more than 90%) using FCD with a sampling time less than 1 minute. However, the accuracy of the constructed trajectories decreased with the sampling time interval. Therefore, a trade-off between the accuracy and operational cost might be considered in the floating car system, depending on the purpose for which it is to be applied.

Some further extensions are envisioned for the $MDP-MM$ algorithm proposed here. In this study, probe vehicles were assumed to travel on the shortest path between two GPS points. In reality, travellers may choose the optimal path based on multiple criteria, such as travel time, number of turns and so on (Li et al. 2011c, Ramaekers et al. 2013). The incorporation of travellers’ route choice behaviour is suggested as an interesting extension of the proposed algorithm, potentially improving the MM accuracy when the sampling time interval is large.
In the cases studies, the MDP-MM algorithm used only a single processor. How to best incorporate the cloud and parallel computing technologies into the MDP-MM algorithm needs further investigation (Li et al. 2011a).

Acknowledgments
The authors are thankful to the anonymous referees for their comments and suggestions that improved the quality of this paper. The work described in this paper was jointly supported by Key Program of National Science Foundation of China (41231171), and research grants from National High-tech R&D Program of China (2012AA12A211), National Science Foundation of China (41201466, 41071285, 41021061), China Postdoctoral Science Foundation (2012M521469, 2013T60741), Shenzhen Scientific Research and Development Funding Program (ZDSY20121019111146499), Shenzhen Dedicated Funding of Strategic Emerging Industry Development Program (JCYJ2012101911128765) and the Research Grant Council of the Hong Kong Special Administration Region (PolyU 5196/10E).

References


