Proceedings of the Federated Conference on Computer Science and Information Systems pp. 563–566 DOI: 10.15439/2020F139 ISSN 2300-5963 ACSIS, Vol. 21

Map Matching Algorithm Based on Dynamic Programming Approach

Alexander Yumaganov
Samara National Research University
Samara, Russia
yumagan@gmail.com

Anton Agafonov
Samara National Research University
Samara, Russia
Email: ant.agafonov@gmail.com

Vladislav Myasnikov
Samara National Research University
Samara, Russia
Email: vmyas@geosamara.ru

***Abstract*—GPS sensors embedded in almost all mobile devices and vehicles generate a large amount of data that can be used in both practical applications and transportation research. Despite the high accuracy of location measurements in 3-5 meters on average, this data can not be used for practical use without preprocessing. The preprocessing step that is needed to identify the correct path as a sequence of road segments by a series of location measurements and road network data is called map matching. In this paper, we consider the offline map matching problem in which the whole trajectory is processed after it has been collected. We propose a map matching algorithm based on a dynamic programming approach. The experimental studies on the dataset collected in Samara, Russia, showed that the proposed algorithm outperforms other comparable algorithms in terms of accuracy.**

I. Introduction

WIDESPREAD deployment of Global Positioning System (GPS) provides a large amount of data describing movement trajectories of pedestrians, bicycles, vehicles, etc. The trajectories are observed as a sequence of GPS records. Each record usually contains ID, latitude and longitude of the GPS sensor and timestamp of the record. GPS sensors usually provide location data with high accuracy up to 5 meters on average, but in some cases, the measurement errors can be much higher, especially in the urban environment. In any case, to use the GPS data in many practical applications and transportation research we first need to perform the preprocessing step that is called the map matching process. Map matching algorithms are applied to identify the correct path as a sequence of road segments by a series of location measurements (GPS records) and road network data. Processed trajectories are an important data source for intelligent transportation systems that can be in such applications as traffic estimation and prediction [1], [2], traffic modelling [3], developing navigation services, user preferences elicitation and training of transportation recommendation systems [4], [5], and so on.

As mentioned earlier, the GPS-trajectories have measurement errors because of multiple factors: atmospheric phenomena, interference from ground-based radio sources, high-rise urban development, vegetation, imperfect hardware and the embedded processing algorithms, and others. We considered a large number of GPS tracks collected by several mobile devices and identify several typical errors:

1) Large geolocation error. The GPS sensor gives several records with the measurement error significantly higher than the usual 3-5 meters.
2) Large time gaps. The GPS sensor does not provide any data for a long time. There are gaps in the track in several minutes or more between high-quality recorded fragments.
3) Continuous deviations from the ground truth path. The GPS sensor for a long time provides coordinates with a low error, the track line looks smooth, however, the deviation of the track line from the true path is several times higher than the average measurement error.
4) Loops when stopped after fast movement.

Given the above measurement errors, the map matching process can be quite challenging. As a result, a number of map matching algorithms have been developed to solve this problem. The algorithms can be categorized by different criteria. In this paper, we consider the online/offline classification. Online map matching methods [6], [7], [8] process positions when the trajectory is still collecting. Offline map matching methods [9], [10] compute the path after the whole trajectory has been collected. In this paper, we consider the offline map matching problem.

Classification and comparative study of map matching algorithms were presented in [11]. In [12], the authors reviewed existing map matching algorithms with the aim of highlighting their qualities, unresolved issues, and provide directions for future studies. The algorithms were compared with respect to positioning sensors, map qualities, assumptions, and accuracy.

In [13], the authors developed a topological point-to-curve map matching algorithm integrated with a Kalman filter. A local incremental algorithm that matches consecutive portions of the trajectory to the road network was proposed in [14].

Weighted-based topological map matching algorithms was proposed in [15], [16]. In [16], the authors integrated raw measurements from GPS, dead-reckoning sensors, and a digital elevation model using an extended Kalman filter in order to increase the accuracy of the map matching process.

In later works, advanced map matching algorithms was proposed. In [17], [18], the authors discussed the possibility of applying Hidden Markov models (HMM) in map matching

The work was partially supported by RFBR research projects nos. 18-07-00605 A, 18-29-03135-mk

algorithms. In the proposed methods, the authors used HMM to find the most likely road route taking into account the measurement noise and the layout of the road network. In [19], the authors proposed a feature-based map matching algorithm that estimates the cost of a candidate path based on both GPS observations and a behavioral model. In [9], the authors presented a map matching algorithm based on Dijkstra's shortest path method that is applicable for large scale datasets. The authors focused on reducing the computational complexity of the algorithm. In [20], the authors also concentrated on designing efficient and scalable map-matching algorithms. They presented an algorithm integrating hidden Markov model with precomputations techniques.

Despite the large number of papers devoted to the map matching problem, the proposed solutions do not allow achieving high accuracy or, in some cases, can not find the correct path at all. In this paper, we focus on developing the map matching algorithm that allows us to identify the correct path with high accuracy. The proposed algorithm consists of two steps: calculating the shortest paths and estimating the paths using a dynamic programming approach.

The paper is organized as follows. In Section II, the main notation and problem statement given. The proposed map matching algorithm with the dynamic algorithm of the shortest path assessment are presented in Section III. Section IV describes the experimental setup and results of experimental studies. Finally, we give a conclusion and possible directions for further research.

II. Problem Statement

A road network is represented as a directed graph $G = (V, W)$, where V is the set of nodes that represent road intersections, W is the set of edges denote road segments. Each node $v \in V$ has the coordinates $\overline{x}_v = (x_v, y_v)$. Each edge $w_{ij} \in W, i, j \in V$ is described by the tuple:

$$w_{ij} = \left(l^w, v^w_{max}, X^w\right),$$

where l^w is a length of the road segment w, v^w_{max} is the maximum allowed speed, X^w is the geometry of the road segment w presented as a set of points.

Define a GPS trajectory as the set of GPS records obtained during the observation:

$$\{\overline{x}_i, t_i\}_{i=\overline{0,I-1}},$$

where I is the number of GPS records, $\overline{x}_i = (x_i, y_i)$ is the coordinates of the tracked objects (latitude and longitude), t_i is the timestamp of i-th GPS record.

Define the ground truth path P as the sequence of the edges (road segments) that was traversed by the vehicle during the observation.

Given the introduced notation, the map matching problem can be formulated as follows:

Given a graph $G = (V, W)$ and a GPS trajectory $\{\overline{x}_i, t_i\}_{i=\overline{0,I-1}}$ find the ground truth path P traversed by a vehicle in a road network.

III. Proposed Approach

A. Map Matching Algorithm

The proposed map matching algorithm can be described as a sequence of the following steps:

1) Determine the start and end nodes by the coordinates of first and last GPS records.
2) For all edges $w \in W$ located at a distance to the GPS records not exceeding $R = 100$ meters, the edge weight is set using the following equation:

$$\varphi(w) = \left(1 - \frac{1}{K}\sum_{k=0}^{K-1} \exp\left(-\alpha \|\overline{x}_k - \overline{x}^p_k(w)\|^2\right)\right) l^w$$

where K is the number of GPS records matched with the edge w, $\overline{x}_k$ are the coordinates of the matched GPS record, $\overline{x}^p_k$ is the coordinates of the GPS record projection on the edge w, α is the coefficient.
For not matched edges the weight is set as follows:

$$\varphi(w) = \beta l^w,$$

where $\beta = 10$ is the coefficient.
3) In the graph with the edge weights set as described above, the shortest path is searched from the start to end node corresponding to the first and last GPS records.
4) The found shortest path is estimated using a dynamic algorithm described in subsection III-B.
5) The algorithm for sequential removal of edges from the path is performed.
Input data: path, assessment of the path, graph. For an edge from the list of path edges:
   a) The edge is removed from the graph.
   b) The shortest path search from the start node of the removed edge to the end node is performed. If there is no such path in the graph, go to step e).
   c) The resulting path is estimated using a dynamic algorithm.
   d) If the resulting assessment is greater than the assessment of the original path, then save the resulting path to the list of best paths;
   e) Restore the deleted edge and move on to process the next edge.

   Output data: the path from the list of best paths, select the path with the maximum assessment value, or an empty path if the list of best paths is empty.
6) The algorithm for sequential removal of edges runs in a loop until an empty path is returned. The last non-empty path will be a solution of the map matching algorithm.

In order to reduce the impact of typical errors in the GPS records, the path assessment in the algorithm for sequential removal of edges is calculated by the following quality criterion:

$$J^* = \begin{cases} J_{p\_cur} - \gamma \frac{l_{p\_cur} - l_{p\_base}}{J_{p\_cur} - J_{p\_base}}, J_{p\_cur} - J_{p\_base} > 0; \\ J_{p\_cur}, \mathit{otherwise}, \end{cases}$$

where J_{p_cur} is the current path assessment, J_{p_cur} is the base path assessment, l_{p_cur} is the current path length, l_{p_base} is the base path length, γ is an empirically selected coefficient.

In the next subsection, we describe the algorithm for the reconstructed path estimation.

B. Dynamic Algorithm for Reconstructed Path Estimation

As a criterion for the reconstruction quality, we use the following:

$$J_p = \sum_{i=0}^{I-1} \exp\left(-\alpha \|\overline{x}_i - \overline{x}_i^p\|^2\right).$$

We need to match the points $\{\overline{x}_i, t_i\}_{i=\overline{0,I-1}}$ with the path p. Firstly, we discretize the path into N points with the discretization step $\Delta = 2$ meters: $p(n) = \overline{x}_n^p, p = \overline{0, N-1}$. Next, we calculate I arrays of proximity similarities between the point $\overline{x}_i$ and the path p as:

$$\varphi_i(n) = \exp\left(-\alpha \|\overline{x}_i - p(n)\|^2\right).$$

The optimization problem is to find the sequence

$$n(i)_{i=\overline{0,I-1}} : \sum_{i=0}^{I-1} \varphi\left(n\left(i\right)\right) \to \max.$$

The main recurrence relation (for the dynamic programming algorithm) has the following form:

$$\max_{n(i)} \sum_{i=0}^{I-1} \varphi_i\left(n\left(i\right)\right) = \max_{n(i_l)=\overline{n(i_{l-1}),N}} \left[\varphi_i\left(n\left(i\right)\right) + \right.$$
$$\left. + \max_{n(i)\leq n(i_l)} \sum_{i=0}^{i_l-1} \varphi_i\left(n\left(i\right)\right) + \max_{n(i)\geq n(i_l)} \sum_{i=i_l-1}^{I-1} \varphi_i\left(n\left(i\right)\right)\right].$$

Introduce the additional notations. Let $\widetilde{\varphi_i}(n)$ be the maximum integral similarity:

$$\widetilde{\varphi_j}(n) = \max_{n(i):i\leq j} \sum_{i=0}^{j} \varphi_i\left(n\left(i\right)\right).$$

Let $\pi_i(n)$ be the list of point positions.

The dynamic programming algorithm to solve the recurrence relation can be described as follows (Algorithm 1).

The result path assessment and the list of point positions are stored in $\widetilde{\varphi_0}(0)$ and $\pi_0(0)$. Using this path assessment, the best path by the specified criteria will be selected in the map matching algorithm.

Algorithm 1 Path estimation algorithm

```
for $i = I-1, 0$ do
  for $n = N-1, 0$ do
    if $i == I-1$ then
      if $n == N-1$ then
        $\widetilde{\varphi_i}(N-1) = \varphi_i(N-1); \pi_i(N-1) = \{N-1\}$
      else
        if $\varphi_i(n) > \widetilde{\varphi_i}(n+1)$ then
          $\widetilde{\varphi_i}(n) = \varphi_i(n); \pi_i(n) = \{N-1\}$
        else
          $\widetilde{\varphi_i}(n) = \widetilde{\varphi_i}(n+1); \pi_i(n) = \pi_i(n+1)$
        end if
      end if
    else
      if $n == N-1$ then
        $\widetilde{\varphi_i}(N-1) = \varphi_i(N-1) + \widetilde{\varphi_{i+1}}(N-1)$
        $\pi_i(N-1) = \pi_{i+1}(N-1); \pi_i(N-1).add(N-1)$
      else
        if $\varphi_i(n) + \widetilde{\varphi_{i+1}}(n) > \widetilde{\varphi_i}(n+1)$ then
          $\widetilde{\varphi_i}(n) = \varphi_i(n) + \widetilde{\varphi_{i+1}}(n)$
          $\pi_i(n) = copy(\pi_{i+1}(n));\ \pi_i(n).add(n)$
        else
          $\widetilde{\varphi_i}(n) = \widetilde{\varphi_i}(n+1); \pi_i(n) = \pi_i(n+1)$
        end if
      end if
    end if
  end for
end for
```

IV. Experiments

Experimental studies of the map matching algorithms were carried out for a large-scale transportation network of Samara, Russia, consisting of 47274 road segments (edges) and 18582 nodes. As a source dataset, we used 20 manually collected tracks recorded by two mobile devices.

We compare the proposed dynamic-based algorithm (DBA) with the FMM algorithm [20] and the HMM-based algorithm [17] implemented as a part of the GraphHopper library [21].

To evaluate the map matching accuracy, we used two metrics: Route Mismatch Fraction (RMF) introduced in [17] and the accuracy metric (A) was used in [20].

The RMF is computed as:

$$RMF = \frac{1}{M} \sum_{m=0}^{M-1} \frac{l_{gt}^m}{l_+^m + l_-^m},$$

where M is the number of tracks, l_{gt}^m is the m-th ground truth path length, l_+ is the length of the road segments in the m-th matched path that are not in the m-th ground truth path (erroneously added), l_- is the length of the road segments in the m-th ground truth path that are not presented in the m-th matched path (erroneously subtracted).

The accuracy metric is the average of the overlapping ratio between the ground truth path (GT) and the matched path(MP):

$$Accuracy = \frac{1}{M} \sum_{m=0}^{M-1} \frac{|GT[m] \cap MP[m]|}{|GT[m] \cup MP[m]|}.$$

Table I presents a comparison of the accuracy of the selected map matching algorithms by the described criteria.

The accuracy of the proposed algorithm is higher than the accuracy of the baseline map matching algorithms.

To visually evaluate the quality of the algorithms, ground truth paths and matched paths were displayed on the map.

TABLE I
ACCURACY MEASUREMENTS OF MAP MATCHING ALGORITHMS

	RMF	**Accuracy**
DBA	**0.135**	**0.876**
FMM	0.245	0.836
HMM	0.744	0.43

Fig. 1 shows an example of the map matching result. The FMM matched path is shown by the green line, the DBA matched path is shown by the dash black line.

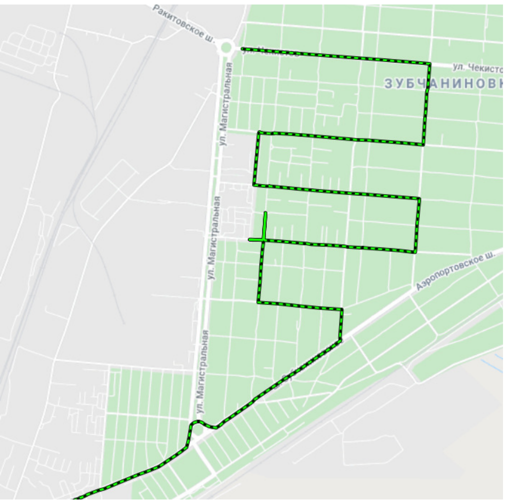

Fig. 1. Example of the map matching result

As can be seen from the picture, both algorithms provide good results, but the FMM algorithm sometimes has large errors on intersections.

V. Conclusion

In this paper, we consider the offline map matching problem in which the whole trajectory is processed after it has been collected. The proposed algorithm consists of two steps performed in a cycle: path estimation and sequential removal of edges from the path. To estimate the matching path, we presented a map matching algorithm based on a dynamic programming approach. Experimental studies conducted on manually collected tracks in Samara, Russia, allow us to conclude that the proposed algorithm has high accuracy and superior other baseline methods by selected criteria.

In future studies, we will investigate our algorithm on publicly-available datasets and focus on improving the computational efficiency of the algorithm.

References

[1] A. Agafonov and A. Yumaganov, "Short-term traffic flow forecasting using a distributed spatial-temporal k nearest neighbors model," in *Proceedings - 21st IEEE International Conference on Computational Science and Engineering, CSE 2018*, 2018. doi: 10.1109/CSE.2018.00019 pp. 91–98.

[2] A. Nagy and V. Simon, "Identifying hidden influences of traffic incidents' effect in smart cities," in *Proceedings of the 2018 Federated Conference on Computer Science and Information Systems, FedCSIS 2018*, 2018. doi: 10.15439/2018F194 pp. 651–658.

[3] Y. Amara, A. Amamra, Y. Daheur, and L. Saichi, "A GIS data realistic road generation approach for traffic simulation," in *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019*, 2019. doi: 10.15439/2019F223 pp. 385–390.

[4] D. Da Silva, J. Torres, A. Pinheiro, F. De Caldas Filho, F. Mendonca, B. Praciano, G. De Oliveira Kfouri, and R. De Sousa, "Inference of driver behavior using correlated IoT data from the vehicle telemetry and the driver mobile phone," in *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019*, 2019. doi: 10.15439/2019F263 pp. 487–491.

[5] V. Myasnikov, "Reconstruction of functions and digital images using sign representations," *Computer Optics*, vol. 43, no. 6, pp. 1041–1052, 2019. doi: 10.18287/2412-6179-2019-43-6-1041-1052

[6] M. Kubička, A. Cela, H. Mounier, and S. Niculescu, "On designing robust real-time map-matching algorithms," in *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, 2014. doi: 10.1109/ITSC.2014.6957733 pp. 464–470.

[7] C. White, D. Bernstein, and A. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1-6, pp. 91–108, 2000. doi: 10.1016/S0968-090X(00)00026-7

[8] H. Wei, Y. Wang, G. Forman, Y. Zhu, and H. Guan, "Fast Viterbi map matching with tunable weight functions," in *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 2012. doi: 10.1145/2424321.2424430 pp. 613–616.

[9] D. Fiedler, M. Čáp, J. Nykl, P. Žilecký, and M. Schaefer, "Map Matching Algorithm for Large-scale Datasets," *arXiv:1910.05312 [cs, eess]*, Sep. 2019, arXiv: 1910.05312. [Online]. Available: http://arxiv.org/abs/1910.05312

[10] Y. Li, Q. Huang, M. Kerber, L. Zhang, and L. Guibas, "Large-scale joint map matching of GPS traces," in *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 2013. doi: 10.1145/2525314.2525333 pp. 214–223.

[11] M. Kubicka, A. Cela, H. Mounier, and S.-I. Niculescu, "Comparative Study and Application-Oriented Classification of Vehicular Map-Matching Methods," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 150–166, 2018. doi: 10.1109/MITS.2018.2806630 Conference Name: IEEE Intelligent Transportation Systems Magazine.

[12] M. Hashemi and H. A. Karimi, "A critical review of real-time map-matching algorithms: Current issues and future directions," *Computers, Environment and Urban Systems*, vol. 48, pp. 153–165, Nov. 2014. doi: 10.1016/j.compenvurbsys.2014.07.009

[13] D. Srinivasan, R. Cheu, and C. Tan, "Development of an improved ERP system using GPS and AI techniques," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, vol. 1, 2003. doi: 10.1109/ITSC.2003.1252014 pp. 554–559.

[14] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, "On map-matching vehicle tracking data," in *VLDB 2005 - Proceedings of 31st International Conference on Very Large Data Bases*, vol. 2, 2005, pp. 853–864.

[15] H. Yin and O. Wolfson, "A weight-based map matching method in moving objects databases," in *Proceedings of the International Conference on Scientific and Statistical Database Management, SSDBM*, vol. 16, 2004, pp. 437–438.

[16] L. Li, M. Quddus, and L. Zhao, "High accuracy tightly-coupled integrity monitoring algorithm for map-matching," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 13–26, Nov. 2013. doi: 10.1016/j.trc.2013.07.009

[17] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 2009. doi: 10.1145/1653771.1653818 pp. 336–343.

[18] R. Raymond, T. Morimura, T. Osogami, and N. Hirosue, "Map matching with Hidden Markov Model on sampled road network," in *Proceedings - International Conference on Pattern Recognition*, 2012, pp. 2242–2245.

[19] Y. Yin, R. Shah, and R. Zimmermann, "A general feature-based map matching framework with trajectory simplification," in *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS 2016*, 2016. doi: 10.1145/3003421.3003426

[20] C. Yang and G. Gidófalvi, "Fast map matching, an algorithm integrating hidden Markov model with precomputation," *International Journal of Geographical Information Science*, vol. 32, no. 3, pp. 547–570, 2018. doi: 10.1080/13658816.2017.1400548

[21] "GraphHopper library," Jul. 2020, original-date: 2014-12-10T09:38:00Z. [Online]. Available: https://github.com/graphhopper/map-matching