2015 IEEE Intelligent Vehicles Symposium (IV)
June 28 - July 1, 2015. COEX, Seoul, Korea

Dataset for testing and training of map-matching algorithms

Matěj Kubička, Arben Cela, Philippe Moulin, Hugues Mounier, and S.I. Niculescu

***Abstract*— We present a large-scale dataset for testing, benchmarking, and offline learning of map-matching algorithms. For the first time, a large enough dataset is available to prove or disprove map-matching hypotheses on a world-wide scale. There are several hundred map-matching algorithms published in literature, each tested only on a limited scale due to difficulties in collecting truly large scale data. Our contribution aims to provide a convenient gold standard to compare various map-matching algorithms between each other. Moreover, as many state-of-the-art map-matching algorithms are based on techniques that require offline learning, our dataset can be readily used as the training set. Because of the global coverage of our dataset, learning does not have to be be biased to the part of the world where the algorithm was tested.**

I. INTRODUCTION

Many map-matching algorithms have been published so far, but there is no standard methodology to estimate their performance. Many authors test their contributions only on simulations. Those who perform field testing most often commit tests that are limited in size, usually without comparison to other algorithms. Reasons for this are twofold: first, it is not immediately clear how to estimate performance of such algorithm (as mentioned in an early paper by White et al. [4]) and secondly, until recently it was cost prohibitive to collect a large-enough dataset for robust testing.

A. Problem statement

We are provided with a *track* and a *map* and we wish to obtain a *route*. A track is a finite, ordered set of *geopoints*, where each geopoint has an assigned position on Earth and a timestamp. A map is modeled as a directed graph consisting of two sets of nodes and arcs. Nodes have assigned position on Earth and arcs represent linear road segments between two nodes.

Note that our definition of a map is the simplest representation of a road network. In particular, such graph embeds curvature of streets as well as one-way restrictions in its topology. Some authors use slightly different model where each arc is a curved street with its shape encoded as attributes.

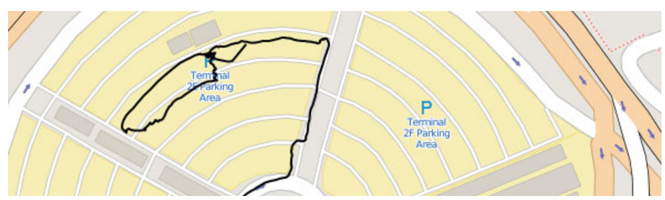
A *route* is a contiguous sequence of arcs in a map on which a vehicle is traveling. The *map-matching* problem deals with matching a track to a map in order to obtain a route. Specifically, we match a track to a sequence of contiguous arcs on a map. As each arc represents a part of a street, our goal is to recover a sequence of streets on which the vehicle travels. This task is essential for a variety of problems such as routing, location aware services, and floating car data systems.

The incongruence between the track and the route is often referred to as "*spatial mismatch*". Map-matching is then a method to correct this mismatch.

(a) problematic track feature example

(b) positioning system error example

(c) map error example



(d) parking lot

Fig. 1. examples of problematic situations

B. Motivation

The main motivation for this work stems from our inability to properly investigate properties of our own algorithm [3]. The approach we took was based on a small set of test runs in rural and urban areas with a predetermined route. We have ran our algorithm on tracks collected on these test runs and observed that the resulting route was matched correctly. It

Matěj Kubička, Hugues Mounier and S.I. Niculescu are with Laboratoire des Signaux et Systemes, CNRS/Supelec, 91192, Gif-Sur-Yvettes Cedex, France. matej.kubicka@lss.supelec.fr, hugues.mounier@lss.supelec.fr, silviu.niculescu@lss.supelec.fr.

Arben Cela is with UPE, ESIEE Paris, 93162, Noisy-Le-Grand Cedex, France, arben.cela@esiee.fr

Philippe Moulin is with IFP Energies nouvelles, 1 & 4, avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex - France, philippe.moulin@ifpen.fr

978-1-4673-7266-4/15/$31.00 ©2015 IEEE

would have been tempting to state that our algorithm was flawless; however, these tests were limited in size and we were aware that they may lack scenarios with the potential of diminishing the performance of our algorithm. An exhaustive search of current literature failed to uncover a standard map-matching benchmark to compare our algorithm to.

C. Map-matching background

Errors in the output of a map-matching algorithm can be broadly categorized as follows:

- *Positioning system errors* are due to measurement noise and temporal gaps in data caused by satellite signal loss.
- *Map modeling errors* are caused by land features that do not fit map modeling assumptions.
- *Map data errors* are due to missing or misplaced arcs or due to errors in their attributes (speed limits, turn restrictions, etc..)
- *Algorithm errors* refer to situations that can diminish performance of the algorithm due to its design or implementation.

Algorithm errors refer to map-matching itself, while map and positioning system errors come from underlying subsystems that feed the map-matcher with information.

See Figure 1 for examples. Figure 1a shows a hive, which is observed when the vehicle is stationary. Some designs of map-matching algorithms tend to match arcs incorrectly in close by area or even wander away out of the area.

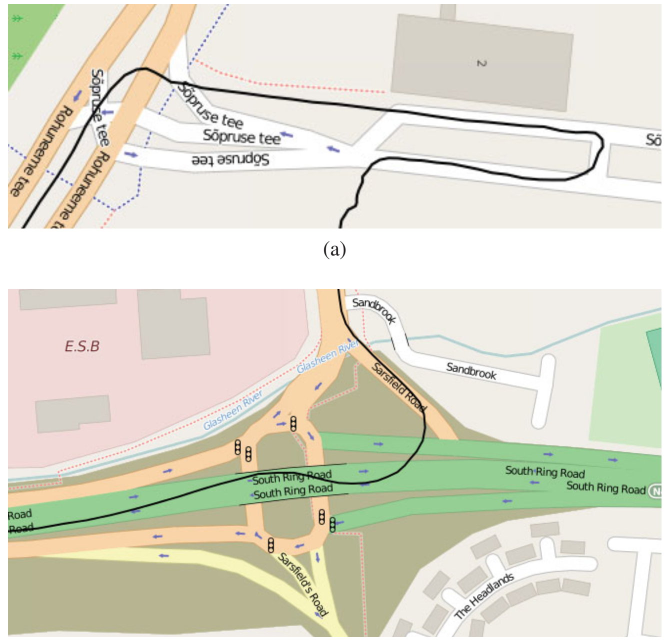
Figure 1b shows an area with extensive positioning system errors, presumably due to bad satellite visibility, non-line-of-sight, or multipath errors. Satellite navigation systems do not fully guarantee positioning precision even when integrity and horizontal protection limits are considered.

Figure 1c shows a missing arc on a map. This type of situation is common for private property and parking lots. If the arc is missing, then this error can be easily corrected, but if there is actually a parking lot in that area, then it cannot be modeled as a graph and some other workaround will need to take place instead.

Typical remedy for the problem of parking lot representation is to map the shape of its corridors as on Figure 1d. This generates particularly tough scenarios for map-matching as the algorithm is forced to map-match specific position on area so condensed that arcs in the map are often closer to each other than best-case resolution of the positioning system.

Two important consequences of the inability to guarantee information from the map and the positioning system are:

1) *Evaluating map-matching performance with respect to reality is not objective*. No map-matching algorithm can guarantee its performance as it is conditioned on the quality of information provided to it. Hence, any objective metric of map-matching performance must decouple map-matching algorithm errors from map and positioning errors.
2) *Map and positioning errors are partially observable*. It is possible to observe incongruences between the map and the track to identify error conditions (i.e. for integrity monitoring). For examples see Figure 2. However, there is no guarantee that these incongruences will emerge in all errorneous scenarios. Mutual errors from map and positioning system can, theoretically, create seemingly congruent scenarios even they are errorneous with respect to the reality.



(a)

(b)

Fig. 2. Examples of incongruent situations

D. Paper organization

The next section is concerned with related work. In Section III we introduce techniques we used to create the dataset and also mention some precision limitations imposed by the nature of the map-matching problem. In Section IV we describe the organization of the dataset and discuss its contents. We conclude in section V and mention future work in Section VI. The dataset itself is available online at [1].

II. Related work

There have been numerous published map-matching algorithms. Most authors use their own categorization schemes so it is not easy to navigate the plethora of methods available. Please refer to our sibling paper [2] where we provide a thorough survey of current methods. Below is a list of a broad categorization of map-matching applications and a few notes that will be important in the next sections.

- *Indoor/outdoor algorithms* - while outdoor map-matching commonly make use of satellite navigation [5], [4], [7], indoor map-matching has to make use of other navigation methods such as inertial navigation, radio beacons, etc. [11]
- *Pedestrian/vehicular map-matching* - pedestrian map-matching [10] is more difficult than vehicle map-matching as pedestrians can be both indoors and outside. This is, strictly speaking, possible even with vehicular

map-matching, but we usually assume that vehicles are restricted to outdoor movement.

- *Online/offline applications* - online map-matching [4], [3] has to provide a partial route while actively collecting the current track. Typical applications span from personal navigation assistants to floating car data. Some authors use the term real-time map-matching to underline the fact that online map-matching gives a result immediately while offline map-matching runs on the whole track after it has been collected. Offline map-matching is used for surveying applications such as fleet management [5], [7], [8].
- *Low/high sampling rate applications* - low sampling rate applications refer to cases where positioning data arrives in intervals higher than 30 seconds [8], [9]. This number is somewhat arbitrary as different authors consider different thresholds, nonetheless a common denominator in these applications is that identifying the correct path between two consecutive position measurements becomes nontrivial.

Note that this is by no means a complete or necessarily correct categorization, we mention it to give the reader an idea of how general the map-matching problem is and where our dataset fits in this scheme. For a more complete discussion please refer to [2].

The dataset presented here is meant for outdoor, vehicular applications. It can be used to test both offline and online algorithms. We used a sampling rate of 1 Hz on all the tracks so one can easily reduce it by down-sampling and test the algorithm on any integral sampling period higher than 1 second. Note that many tracks still contain temporal gaps due to satellite signal loss. If strictly uniform sampling is required, then it suffices to filter our dataset for records that are not labeled to have gaps (see Section IV).

Authors Newson and Krumm [5] and Zheng et al. [6] give their own testing data sets. The former provided a complete dataset with track, map, and map-matched route of a 3-hour drive in Seattle, WA, USA. The latter contains a large volume of trajectories collected over a period of 3 years in China. Only the tracks are available - map and map-matched routes were not included.

A. Performance measures

White et al. [4] wrote in their early paper on map-matching that *"it is not immediately clear what measures of performance are most appropriate or what scenarios should be evaluated. In an abstract sense, it is clear that we would like the algorithm to perform perfectly when the errors go to zero, but it is not entirely clear what that means in practice."*. This seems to hold true even today as authors do not agree on any particular evaluation method.

Most authors compare performance of their algorithm by counting correctly matched geopoints with respect to total number of geopoints measured [4], [3]. Some authors use derived measures (i.e. [9]) such as counting the number of correctly matched arcs with respect to the total number of arcs in the correct route, or summing the total length of correctly matched arcs and comparing it to the total length of the route.

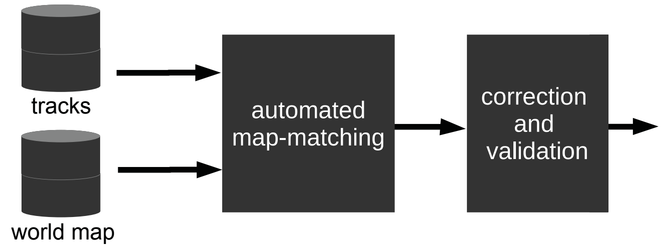


Fig. 4. toolchain used to create the dataset

Further derivatives of these basic approaches are also common. Newson and Krumm [5] counted the length of incorrectly matched arcs with respect to the total length of the correct route, which is reciprocal to the method used by Lou et al. [9].

Another method is used by Quddus et al. [12]. They used double-carrier high-sensitivity GPS receiver integrated with an inertial navigation system as a "superior" positioning system that was assumed to render the map-matching problem trivial. Performance of their map-matching algorithm was measured by observing the degree of agreement between outputs from the superior system and the system under test.

III. Methodology

Tracks featured in our dataset originate from a publicly available collection called Planet GPX [15]. This dataset is part of the OpenStreetMap project [16]. It was collected over a period of nine years by volunteers worldwide for automated route shaping and turn restriction detection in the OpenStreetMap.

Planet GPX features an open license and millions of tracks spread worldwide (for coverage see Figure 3). Therefore, it has potential for map-matching even if it was never intended for this purpose. However, as the data are collected by volunteers with various devices and sometimes buggy software, the dataset contains some unexpected material. There is an unusual amount of tracks with only a single geopoint. Moreover, one can observe that some users successfully uploaded airplane routes and other curiosities, such as boat trip to Antarctica. Hence, we applied a series of filters on the tracks in Planet GPX. First, we removed all tracks that are missing some or all timestamps and all tracks that are not fully causal (meaning that timestamps of successive geopoints in the track are not monotonically increasing). Then we removed all invalid tracks with geopoints on latitudes or longitudes outside their boundaries ($\pm 90°$ and $\pm 180°$ respectively).

These filters allowed us to remove degenerate tracks from the dataset. Other filters helped to enforce specific properties of the tracks in our dataset. First, we removed all tracks with a total length less than 5 kilometers and more than 100 kilometers in order to avoid exceedingly small or exceedingly large tracks. Second, we removed all tracks with sampling periods other than one second.

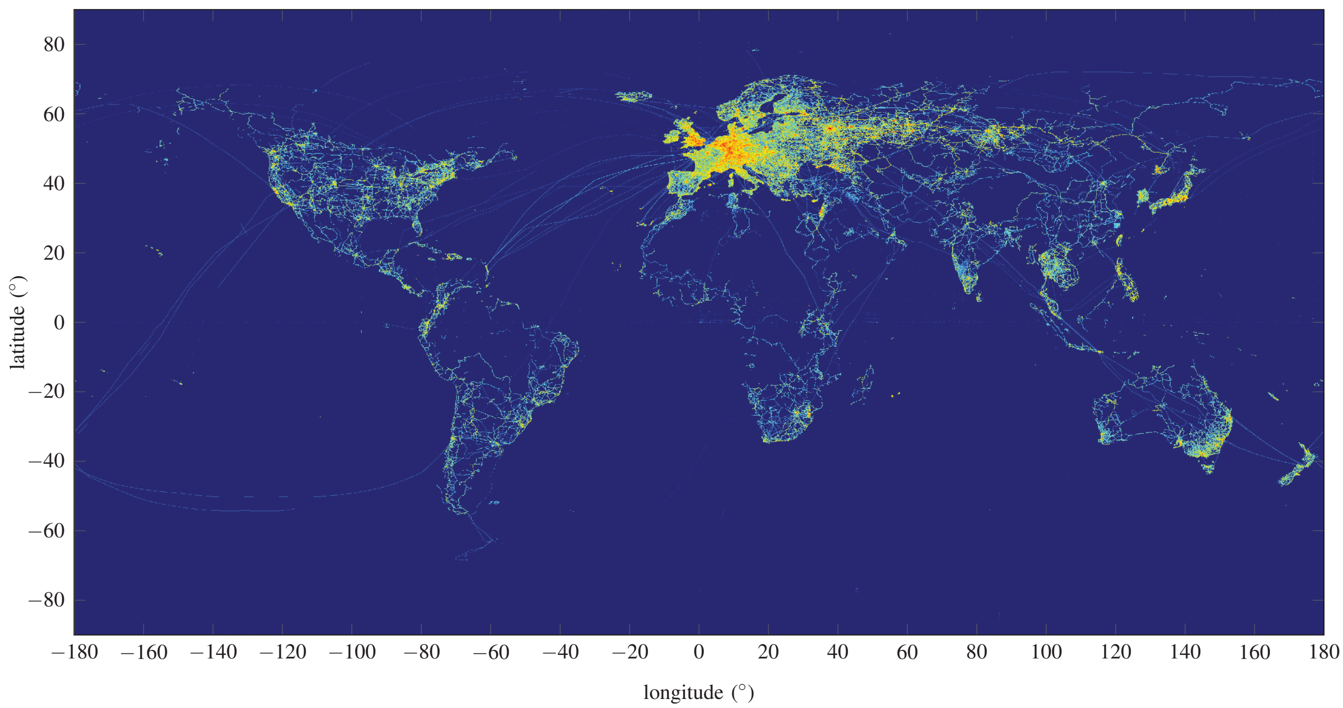


Fig. 3. world-wide intensity plot with distribution of tracks in Planet GPX dataset

In order to estimate the sampling rate, we observed the 20^{th} and the 80^{th} percentile of delays between any two contiguous geopoints. If both values were in the range of 500ms from each other, then we took the median of the delays between samples as the sampling rate. Otherwise, we deemed the track to have an unknown sampling rate.

As mentioned in Section I-A, each track is accompanied with an underlying map and a correctly map-matched route on the map. We make use of the publicly available OpenStreetMap to extract road-network information under each track and use our custom made software to map-match and hand-correct the routes. See Figure 4 for the toolchain used to create the dataset.

A. Precision considerations

Instead of running a world-wide data collection campaign, we used pre-existing data originally collected and meant for other purposes. Since we have not actually driven on these routes, we cannot guarantee absolute correctness of the dataset. When hand-correcting the tracks we implicitly assumed that a human mind is a superior map-matcher. This seemed reasonable at first, but we observed that there are situations where correct map-matching is not clear even to us. Such situations occur when there is not enough information to decide with reasonable level of confidence, as mentioned in Section I-B.

We had two options to deal with this: we could either exclude such problematic tracks from the dataset, or map-match them anyway at the risk that matching would be incorrect. The former would make our dataset biased towards particular brand of tracks that are easier in some sense while the latter would potentially introduce errors into our dataset.

Authors of NCHRP's Synthesis 301 [13] ran a survey among dozens of organizations that use map-matching on a regular basis. On page 27 they state the following: *"The implication of the DOTs (edit: Departments Of Transportation) responses to the spatial mismatch problem indicates that adjusting the data points is as much an art as it is a science"*. We imply that their findings apply to map-matching problem as well since map-matching can be viewed as an algorithm for automated spatial mismatch correction.

Therefore, we chose the second option and provided map-matched tracks even in severely incongruent situations. This decision carries both advantages and disadvantages. On the plus side, it enables learnable algorithms to learn some of the "human intuition" reflected in our matching. On the other side, our data contains situations where correct matching is arbitrary (see Figure 2 for such situations) due to incongruences between a track and a map which leaves little information about the actual route taken by the vehicle.

Note that we have labeled each of these tracks as "severely incongruent" for easy identification (see Section IV). This allows to exclude these tracks from automated tests, if needed.

It is important to stress that our dataset cannot be described as ground-truth data due to these congruence issues that emerge when dealing with map-matching.

IV. The dataset

The dataset (available online at [1]) is provided as a set of *records*. Each record has a unique identifier and contains a track, a map and a correctly map-matched route. Moreover, we have labeled each track with a selection of features that might pose difficulties to some map-matching algorithms:

Fig. 5. distribution of records

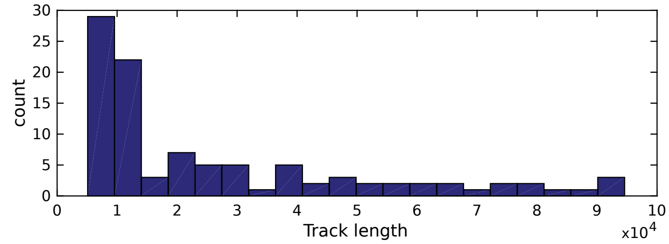


Fig. 6. histogram of track lengths in the dataset

- *u-turns* - the vehicle turned around in the middle of the street
- *hives* - a large volume of geopoints packed in a small area
- *loops* - the vehicle was traveling in circles
- *gaps* - temporal gaps in the track
- *severe congruence issues* - situations where the map and the track are incongruent or dissimilar.

As mentioned above each record contains a track, a map and a corresponding route. These are stored in four separate files - one for the track, one for the route and two for the map. Each file contains a tab-separated human-readable table that can be inspected manually or processed using any programming language[1].

- The *track file* is stored at `<id>/<id>.track` where `<id>` refers to record identifier. It has three columns, longitude, latitude and time. Each line describes a single geopoint. The timestamp of the first geopoint is zero and increases monotonically with proceeding geopoints.
- The *map files* are stored in `<id>/<id>.nodes` and `<id>/<id>.arcs`. As mentioned in section I-A, the map is modeled as a graph consisting of two sets of nodes and arcs. They are stored in the aforementioned files. Nodes are stored as a table of two columns (longitude and latitude) where each row defines a single node. Arcs are stored as a table of two columns, where each row defines single arc and the columns refer to the starting and ending nodes in the nodes table by their (zero-based) row number.
- The *route file* is stored in file `<id>/<id>.route`. This file has a single column with the sequence of arcs on which the vehicle traveled. Each line refers to an arc in the arcs table using its (zero-based) row number.

Records are stored in folders named according to their identifiers and listed in the provided `metadata.xml` file. This file contains a description of all the records. For convenience we also provide `metadata.xsd` file with XML schema definition of the `metadata.xml` file. The structure of `metadata.xml` is self-explanatory, see Listing 1 for an example.

[1]For example in Matlab and Octave one could load the tables as matrices using the `dlmread` command.

The records are listed as child elements of root element `<records>`. Each record contains a list of its features (stored as `<tag>` elements) and reference to the four files with track, map and route. To ensure integrity, it also mentions the MD5 checksum for each file.

Additionally, element `<spatial-mismatch>` gives information about spatial mismatch between the track and the route. An attribute "`mad`" is median absolute deviation and attribute "`max`" is peak distance between the track and the route. The median absolute deviation was introduced by Hampel [14] who attributed it to Gauss. In context of map-matching it was already used by Newson and Krumm [5]. It is a robust measure of variability computed as

$$MAD = \mathrm{med}_i\left(\left|X_i - \mathrm{med}_j(X_j)\right|\right) \tag{1}$$

where X is univariate data set. In order to obtain consistent estimator of standard deviation one has to take

$$\hat{\sigma} = K \cdot MAD \tag{2}$$

where K is scale parameter that depends on probability distribution function. For normal distribution $K \approx 1.4826$, for exponential distribution $K \approx 2.0781$ and for Rayleigh distribution $K \approx 2.230$. These distributions are often used in the literature to characterize spatial mismatch.

```
<?xml version="1.0" encoding="utf-8"?>
<records xsi:noNamespaceSchemaLocation="metadata.
    xsd">
  <record id="0" state="revised">
    <track path="./build/00000000/00000000.track"
      md5="ebf86b337907d9cea4f1e229fb7efd96"/>
    <nodes path="./build/00000000/00000000.nodes"
      md5="cd27fbad5dadfc873ed989213b9f5705"/>
    <arcs path="./build/00000000/00000000.arcs"
      md5="081580e6436cdf829662816601e9e8f0"/>
    <route path="./build/00000000/00000000.route"
      md5="cb0c3bea37bc61bd0e73292c1a9c2b1b"/>
    <spatial-mismatch mad="2.2448" max="61.1719"/>
    <tag key="feature" value="gaps"/>
    <tag key="feature" value="congruence-issues"/>
  </record>
  <record>
    ... next record ...
  </record>
</records>
```

Listing 1. Example of `metadata.xml`

A. Working with the records

We intentionally avoided any complex representations of the data to simplify their use. Specifically, we made sure that the data is human-readable and that it is always stored as simple tables that can be easily loaded by common tools such as Matlab or Microsoft Excel. We provide simple Matlab script `plot_record.m` together with the dataset to facilitate working with the data. The script takes record identifier as input and plots the map together with the track and the route.

B. Contents

The current version of our dataset contain a selection of 100 records. This collection has 73 tracks with gaps, 25 tracks with "U"-turns, 24 tracks with loops, 3 tracks with hives, and 20 records with severe congruence issues.

As mentioned above the track lengths are limited in range from 5 to 100 kilometers, see Figure 6 for histogram of their distribution. All tracks have the same sampling rate of 1 Hz.

In total the dataset contains 247,251 points and 2,695 kilometers of tracks, see Figure 5 for their distribution around the globe.

The dataset also contains scenarios where correct matching is either impossible or arbitrary due to errorneous or insufficient information. Where possible, these tracks are map-matched anyway and labeled as incongruent for easy identification.

V. CONCLUSIONS

Our presented dataset for testing and training of map-matching algorithms features a volume of map-matched tracks from around the world. Until now, there were obstacles that prevented researchers and engineers from collecting large enough dataset for more complete testing, benchmarking, and learning. With 2,695 kilometers of correctly map-matched routes distributed worldwide our dataset helps to overcome this obstacle.

A detailed study of map-matching problems can be facilitated with this. Map-matching algorithm designers can now test and observe the behavior of their algorithm on a rich set of scenarios. Researchers can also use it to better understand the impact of errors in maps and tracks to the map-matching problem.

Most importantly, the dataset can be used to compare performance of different map-matching algorithms. This was not simple until now as discussed in Section I. Our contribution enables this data to be used as the so-called "gold standard".

Similarly, our work can be used as a training set for offline learning. Especially the latest map-matching algorithms are often based on Fuzzy logic, Hidden Markov models, Conditional random fields, Dempster-Shafer theory, etc. These require training before deployment which our data can facilitate.

All the tracks feature 1 Hz sampling rate that can be easily downsampled to any integral sampling rate, however the data are not uniformly sampled due to occasional satellite signal loss. We have labeled tracks with temporal gaps for easy identification.

VI. FUTURE WORK

Our next step will be to use the dataset to evaluate and compare the performance of a selection of state-of-the-art map-matching algorithms. We hope this will provide new insights into the map-matching problem and provide basis of comparison in this field of research.

Future extensions to the dataset, if necessary, will be additive and shall not change the data representation described in this publication.

REFERENCES

[1] Matěj Kubička, Arben Cela, Hugues Mounier, Philippe Moulin and S.I. Niculescu, *"Dataset for map-matching"* [Online]. Available: http://www.matejk.cz/map-matching

[2] Matěj Kubička, Arben Cela, Hugues Mounier, Philippe Moulin and S.I. Niculescu, *"On map-matching"*, IEEE Trans. on ITS, to be published.

[3] Matěj Kubička, Arben Cela, Hugues Mounier, *"On designing robust real-time map-matching algorithms"*, ITSC, 2014, p. 464-470

[4] White, C.E., Bernstein, D., Kornhauser, A.L., 2000. *"Some map-matching algorithms for personal navigation assistants"*. Transportation Research Part C 8, 91-108.

[5] Paul Newson and John Krumm, *"Hidden Markov Map Matching Through Noise and Sparseness"*, 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2009), Seattle, WA, pp. 336-343.

[6] Yu Zheng, Lizhu Zhang, Xing Xie, Wei-Ying Ma. *"Mining interesting locations and travel sequences from GPS trajectories"*. In Proceedings of International conference on World Wild Web (WWW 2009), Madrid Spain. ACM Press: 791-800.

[7] Alt, H. et al., *"Matching Planar Maps"*. Journal of Algorithms, 2003. 49: p. 262283.

[8] Yuan, Y. Zheng, C. Zhang, X. Xie, and G. Sun, *"An interactive-voting based map matching algorithm."* in MDM, 2010, pp. 4352.

[9] Lou, Y.; Zhang, C.; Zheng, Y.; Xie, X.; Wang, W.; Huang, Y; *"Map-Matching for Low-Sampling-Rate GPS Trajectories"*, ACM GIS '09 , November 4-6, 2009. Seattle, WA, USA (c) 2009 ACM.

[10] Ren, Ming (2012) *"Advanced Map Matching Technologies and Techniques for Pedestrian/Wheelchair Navigation"*. Doctoral Dissertation, University of Pittsburgh.

[11] Zhuoling Xiao, Hongkai Wen, Andrew Markham, Niki Trigoni. *"Lightweight map matching for indoor localisation using conditional random fields"*. IPSN'14, Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, p. 131-142, IEEE/ACM, 2014.

[12] Quddus, M.A., Noland, R.B., Ochieng, W.Y., 2006b. *"A high accuracy fuzzy logic-based map-matching algorithm for road transport"*. Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 10 (3), 103-115.

[13] NCHRP, 2002, *"NCHRP Synthesis 301: Collecting, Processing, and Integrating GPS Data into GIS"*. National Academy Press, Washington, D.C.

[14] Hampel, F. R., 1974, *"The Influence Curve and its Role in Robust Estimation"*, Journal of the American Statistical Association, 69, 383-393.

[15] OpenStreetMap contributors, *"Planet GPX"* [Online]. Available: http://planet.openstreetmap.org/gps/

[16] OpenStreetMap contributors, *"OpenStreetMap"* [Online]. Available: http://openstreetmap.org/