Trajectory Annotation by Discovering Driving Patterns

Sobhan Moosavi[†], Behrooz Omidvar-Tehrani[‡], Rajiv Ramnath[†]

[†]The Ohio State University (USA), [‡]University of Grenoble Alpes (France)
{moosavinejaddaryakenari.1,ramnath.6}@osu.edu,behrooz.omidvar-tehrani@univ-grenoble-alpes.fr

ABSTRACT

The ubiquity and variety of available sensors has enabled the collection of voluminous datasets of car trajectories that enable analysts to make sense of driving patterns and behaviors. One approach to obtain driving behaviors is to break a trajectory into its underlying *patterns* and then analyze these patterns (aka segmentation). To validate and improve automated trajectory segmentation algorithms, there is a crucial need for a ground-truth against which to compare the results of the algorithms. To the best of our knowledge, no such publicly available ground-truth of car trajectory annotations exists. In this paper, we introduce a trajectory annotation framework and use it to annotate a real-world dataset of personal car trajectories. Our annotation methodology consists of a crowd-sourcing step followed by a precise process of expert aggregation. Our annotation identifies segment borders, and then labels the segment with its type (e.g. *speed-up*, *turn*, *merge*, etc.). The output of our project is a dataset of annotated car trajectories (DACT), and is publicly available for use by the spatiotemporal research community at https://goo.gl/XgsxyJ.

CCS CONCEPTS

• **Information systems** → **Data mining**; *Spatial-temporal systems*;

ACM Reference format:
Sobhan Moosavi[†], Behrooz Omidvar-Tehrani[‡], Rajiv Ramnath[†]. 2017. Trajectory Annotation by Discovering Driving Patterns. In *Proceedings of UrbanGIS'17:3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics, Redondo Beach, CA, USA, November 7–10, 2017 (UrbanGIS'17)*, 4 pages.
https://doi.org/10.1145/3152178.3152184

1 INTRODUCTION

The ubiquity of a variety of sensors (accelerometer, barometer, GPS, etc.) has enabled the collection of voluminous datasets of car trajectories. Examples of such datasets are the New York Taxicab dataset [1], the Porto Taxicab dataset [13], and GeoLife [19]. Such datasets enable analysts to extract insights on driving patterns and behaviors. This has many applications in urban management such as city traffic planning and improving driver safety by reducing the risk of accidents.

Each trajectory in a dataset of car trajectories is a time-stamped sequence of data points described using attributes such as speed, bearing and location. To understand driving behaviors, one approach is to break each trajectory into its underlying patterns, or segments (Figure 1), and then describing the trajectory in terms of the patterns exhibited in these segments [2, 3, 6]. For example, our work in [12] describes a statistical modeling approach to transform a trajectory into a signal and then applying a dynamic programming-based segmentation algorithm to identify the most interesting segments using Minimum Description Length (MDL).

UrbanGIS'17, November 7–10, 2017, Redondo Beach, CA, USA

ACM ISBN 978-1-4503-5495-0/17/11...$15.00
https://doi.org/10.1145/3152178.3152184

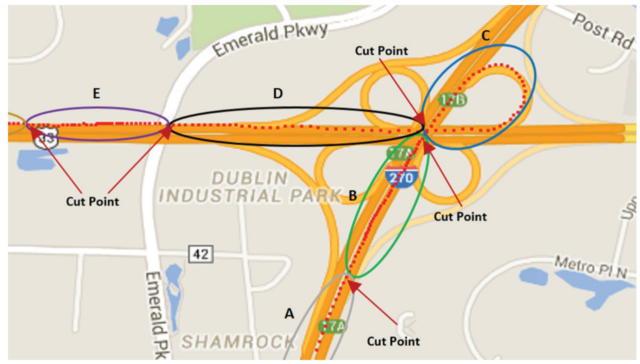


Figure 1: A sample trajectory with several underlying driving patterns specified by ovals. Red arrows show the points of transition between patterns, where new patterns commence.

One of the main shortcomings of all the aforementioned approaches is the lack of real-world experimental studies. The applicability and performance is often validated just through few realistic use cases. There should be a set of *labeled trajectories* as ground-truth in which the true segment borders are specified by domain experts. To the best of our knowledge, there is no such dataset publicly available for the research community. In this paper, we describe and provide a link to such a dataset. We also present methodology for trajectory annotation comprising two main steps, Expert Annotation and Annotation Aggregation.

Contributions. We propose a simple yet effective two-step approach for annotating car trajectories. We apply our approach on a real-world car trajectory dataset and discuss the pains and gains of the process. Last we make the resulting annotation dataset publicly available for spatiotemporal research community.

Outline. First, we review related work in Section 2. Then, we introduce our trajectory annotation framework in Section 3. Next we discuss results of annotating a real-world dataset in Section 4. We introduce the resulting dataset of trajectory annotations in Section 5. Last, we conclude in Section 6.

2 RELATED WORK

This work is an example of spatiotemporal crowdsourcing task as studied before in [8, 15, 16]. The major difference between our proposed annotation framework and other existing approaches is two-fold. First, we recognize that trajectory annotation is highly *subjective*. While there is close consensus when Human Intelligence Tasks (HIT) are used to identify static attributes of a trajectory notions, e.g., "where a street begins", or "where a traffic light is situated", there is considerable disagreement when labelling driving patterns with behavioral labels such as "slow-down" or "jiggling[1]". Generally, it is hard for humans to achieve consensus on "where (in a trajectory) *a change in driving behavior occurs*". Second, our aim is not to just identify how humans think about trajectories, but to come up with gold standard annotations that can be used to validate results of unsupervised and semi-supervised trajectory segmentation algorithms

[1] *Continuously going to left and right while driving.*

(e.g., [2–4, 6, 14]). The specific task of annotation of trajectories (which we formulate in this paper) is also related to prior efforts on specifying *stop* and *move* episodes in trajectories [9, 18]. However, the task of annotation of trajectories is rather more challenging. For instance in our case, each "move" episode may be further annotated by several patterns.

3 ANNOTATION FRAMEWORK

A trajectory T_i in a database of car trajectories $T = \{T_1, T_2, \ldots, T_N\}$ consists of a sequence of data points $\langle p_{i1}, p_{i2}, \ldots, p_{in}\rangle$. Each data point p_{ij} is a tuple of the form $(Time_j, Speed_j, Heading_j, Lat_j\ , Lng_j)$ where $Time_j$ is the timestamp, $Speed_j$ is the vehicle's speed (mph), $Heading_j$ is the direction change based on the previous direction, and Lat_j and Lng_j are the exact location of the vehicle. Segmentation of a trajectory T_i into m segments is the process of discovering a set of cutting indexes $\langle I_{i1}, I_{i2} \ldots, I_{im}\rangle$ which mark the end points[2] of T_i's non-overlapping segments. Note that I_{im} corresponds to the index of the last point of trajectory T_i (i.e., $T_{I_{im}} = T_n$).

The annotation of a trajectory T_i is the task of identifying segment borders (i.e., cutting indexes) within T_i using human expertise. An expert may assign one or more labels to each segment, where examples of labels are *speed-up*, *slow-down*, *turn*, etc. As noted before, our methodology for trajectory annotation contains two steps, Expert Annotation, and Annotation Aggregation. The former refers to a crowdsourcing-style process of assigning trajectories to experts for annotation. The latter is a finalization step which aggregates received annotations of human experts to land on a consensus.

3.1 Expert Annotation

In this step, a human expert u annotates m segments for a trajectory T_i. We assign each trajectory T_i to at least two experts. We prepare a web interface for experts to provide a bird's-eye view on the data under investigation. The trajectory T_i is displayed on a geographical map and can be zoomed-in to see the street view, which enables a more precise annotation. Speed and heading-change profiles are illustrated in form of interactive time-series, where hovering on a data point in one of the time-series, will highlight the same point in all other visualizations (that is, the views are coordinated).

For privacy reasons, we randomly hide some points in T_i so that the characteristics of the driver, car and other entities are not identifiable. For instance, we remove points in the first and last 5 minutes of the trajectory to hide the exact addresses of departure and destination.

During the annotation process, an expert u follows data points in T_i in any of the provided visualizations. When a change is observed at a point p_{ij}, u may decide to declare the end of a segment by clicking p_{ij}. She then may also specify the type of the segment, from the ones listed in Figure 3.

3.2 Annotation Aggregation

High subjectivity in trajectory annotation by humans necessities an aggregation phase. In this step, a human expert e decides to *accept*, *reject* or *refine* each provided annotation for a trajectory T_i in order to finalize the annotation process. A decision is made based on some guidelines and a heuristic baseline. To deal with strong disagreements among annotators, we perform the aggregation in two different modes, Strict and Easy, each of which delivers a distinct set of guidelines.

Strict Aggregation. In the strict mode, our focus is to maximize the usability of *all* experts' annotations. Different experts may identify different segments in a trajectory. In this mode, all the segments are identified as such, whether they are *independent* (e.g., an entire *loop* as in Figure 3) or *dependent* (i.e., segments which occur inside a bigger segment – being part of another segment, e.g., a *slow-down* inside a *loop*, as in Figure 2).

[2] *An end point is also called a cutting point as represented in Figure 1.*

Based on some initial trial-and-error experiments, we identified certain thresholds as follows. The expert e considers any change in the speed larger than $5mph$ as a significant change and marks the segment as either a *speed-up* or *slow-down*. Also, a continuous change in heading values for *five consecutive seconds* counts as a significant change in heading and identifies a segment as *smooth-turn* or *jiggling*. Observing such continuous period ensures that we do not end up with GPS false positives.

Easy Aggregation. In contrast to the strict mode, easy aggregation only identifies independent segments. Also we use larger thresholds for segments. For speed, a change must be by at least $10mph$ to count it as a significant change. For heading, a consecutive series of changes in 10 seconds counts as a segment. Obviously, less segments will be marked in the Easy aggregation comparing to Strict aggregation. Figure 2 provides examples to contrast between easy and strict aggregations.

As a help during decision-making process, the aggregating expert e also has access to the output of a heuristic baseline as a secondary source of information. We build an unsupervised algorithm called AutoAnn (Automatic Annotation) which uses heuristics on speed, heading, and position derived from a pilot study[3] in order to mark segments. By being provided alongside expert annotations, AutoAnn enables experts in the aggregation phase to get to a consensus. We stress that the employed heuristics are very simplistic and cannot replace the human power of decision making. In other words, AutoAnn is only provided as a help to the aggregating expert.

AutoAnn scans each single data point in T_i (in the ascending order of timestamps) and makes comparisons with k neighbor points (experimentally, k is set to 5). A few of the heuristics of AutoAnn is described as follows (see [10] for more details.)

H1. Speed-wise, if a data point p falls into a local maxima among it's k neighbors (i.e., k previous and k subsequent points), then p marks the end of a *speed-up* segment. Similarly, if p falls into a local minimum, p marks the end of a *slow-down* segment.

H2. If a data point p holds as the end of a *slow-down* segment and the speed at p is lower than a *low-speed* threshold (experimentally set to $9mph$), then p marks the end of a *traffic-jam* segment. The intuition is that a *serious* slow-down should be due to a traffic jam.

H3. If the position of a point p is merely identical with the point p' which occurred before p, then p' marks the beginning and p marks the end of a *loop* segment. The intuition is that when a driver performs a loop (for instance through a highway exit as in the top-right plot in Figure 3), the first and last locations of the loop are identical with a tiny error.

H4. Consider the function $d()$ that returns the first-order derivative of heading values. Given points p and p' where p' occurs right after p, the point p marks the end of a *turn* segment iff it satisfies two following conditions: first, the function $d()$ returns 0 for all k neighbors of p, second, $d(p)$ is larger than a heading threshold (experimentally set to 15).

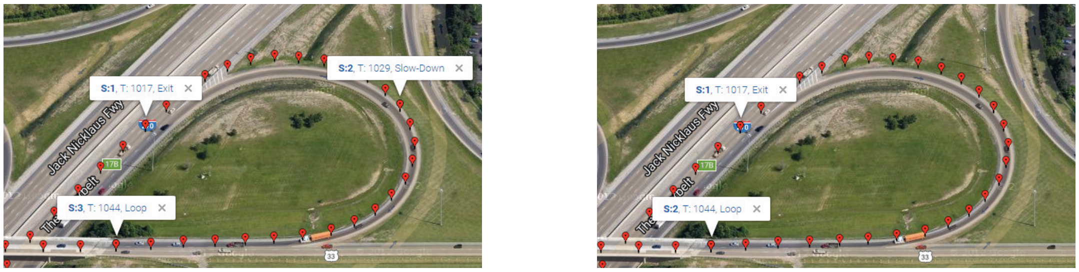
4 ANNOTATION RESULTS

We apply our annotation framework on a real-world dataset of personal car trajectories[4], collected using highly accurate devices connected to On Board Diagnostic (OBD-II) port of vehicles. Note that the data has been collected with a consistent sampling rate of one second, whereas most of the existing public datasets use samples collected at lower rates [1, 13, 19].

A data cleaning phase removes redundant or highly similar trajectories in terms of traffic condition or metropolitan coverage (hence increasing diversity). Table 1 summarizes the final dataset. AutoAnn generates 2418 segments on this data, among which 59% are slow-downs and 20% are traffic-lights.

[3] *Prior to the annotation process, we ran a pilot phase to identify challenges and properties of the task.*

[4] *This dataset is provided by Nationwide Mutual Insurance Company.*



(a) Strict Aggregation **(b) Easy Aggregation**

Figure 2: Difference between strict (a) and easy (b) aggregation: In the first case, we consider both *independent* and *dependent* segments, while, for the second one, we just consider the *independent* segments. Examples of *independent* segments are S1 and S3 in (a) and an example of *dependent* segment is S2 in (a).

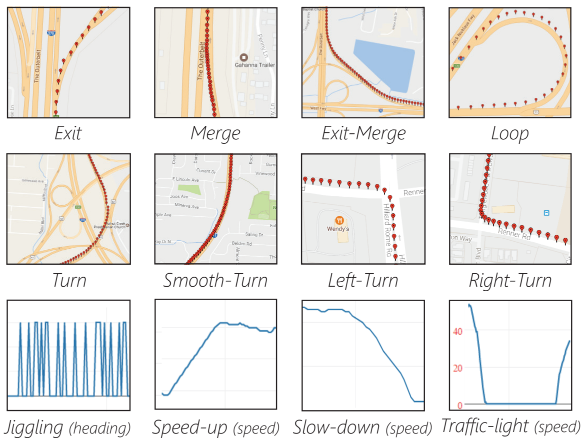


Figure 3: Segment types derived from real data. For the last row, the feature which is used to demonstrate the pattern is shown within parenthesis.

Table 1: Trajectory Dataset to be used for Annotation.

Number of Trajectories	Number of Drivers	Total Driving Time	Average Trajectory Duration
50	19	13.3 *hours*	16 *minutes*

We next present our observations on annotating the dataset of car trajectories. First, we present *time costs*, then analyze *inter-expert agreements*, and finally we discuss the *distribution of annotations*.

Time Cost. Table 2 presents selected statistics from annotation and aggregation phases. We observe that the most time-consuming phase is the strict aggregation where the number of extracted segments is also the largest. Moreover, the annotator experts and the strict-aggregation expert spent almost the same amount of time to annotate a new segment (i.e., 31 seconds). Of course, more segments are generated by the aggregator, which justifies the work of aggregator.

Inter-Expert Agreement. Given two annotator experts u_1 and u_2 for a trajectory T, their set of annotated points is denoted as $ann(u_1, T)$ and $ann(u_2, T)$. We measure the correlation between annotation sets using Jaccard and Cohen's Kappa [7] metrics, defined as follows.

$$Jaccard = \frac{a}{a+b+c}\ , \qquad \kappa = \frac{(\frac{a+d}{a+b+c+d}) - \omega}{1-\omega}$$

Here, $a = |ann(u_1, T) \cap ann(u_2, T)|$, $b = |ann(u_1, T) - ann(u_2, T)|$, $c = |ann(u_2, T) - ann(u_1, T)|$, $d = |T - (ann(u_1, T) \cup ann(u_2, T))|$ and $\omega = ((a + b)(a + c) + (c + d)(b + d))/(a + b + c + d)^2$. Annotation points should not exactly be in the same location to be counted in the variable a. Given a distance approximation threshold τ, two points $p_1 \in ann(u_1, T)$ and $p_2 \in ann(u_2, T)$ are involved in the intersection of annotation sets, iff $Haversine(p_1, p_2) \leq \tau$. Haversine function is a distance metric in spherical space which serves latitudes and longitudes [17].

Figures 4a and 4b show the analysis of inter-expert agreement based on Cohen's Kappa and Jaccard, respectively, by varying the distance threshold τ. Note that for aggregation phases, the agreement analysis is done by comparing the annotations produced by "aggergator" with those generated by the "annotators". Note that the largest agreements are for easy-aggregation. Also, strict-aggregation has a larger agreement than expert-annotation. We summarize few other insights on agreements as follows.

- *On Subjectivity.* We observe that the disagreement between different annotators was large, even with a loose distance threshold of 200 meters. Clearly, trajectory annotation is subjective!
- *On Aggregation.* Figures 4a and 4b illustrate that the aggregator considers the set of annotations of annotator experts in both strict and easy aggregations, and this process boosts the agreement from 44% to about 60% (using Cohen's Kappa metric). In other words, the aggregation reduces disagreements drastically and tackles the subjective nature of the task.
- *Annotation vs. Easy-Aggregation.* We observe that there is a larger agreement between results of easy-aggregation and expert-annotation phases. This is due to exploiting relaxed versions of constraints during the easy-aggregation as opposed to strict-aggregation.
- *Larger Agreements by Cohen's Kappa.* We also observe that Cohen's Kappa tends to return larger agreement values. This is due to the consideration of non-annotated points as well as annotated ones. On the contrary, Jaccard uses only the annotated points.

Distribution of Annotation Types. Figure 5 shows the frequency of segment types in the different phases of annotation. We observe that the most common segment types are *speed-up*, *slow-down*, and *smooth-turn* which cover 70% of strict aggregations, 66% of easy aggregations and 55% of annotations. Interestingly, we also observe a parity in the frequency of correlated segments, such as *exit* with *merge*, *left-turn* with *right-turn*, and *speed-up* with *slow-down*.

5 DACT: ANNOTATION DATASET

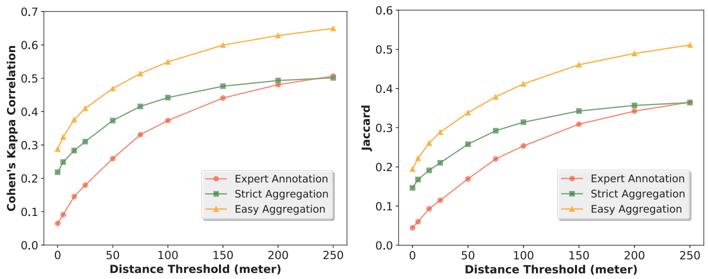
We briefly present the outcome of this study, i.e., a dataset of annotated car trajectories called DACT. The dataset functions as a ground-truth for car trajectory segmentation. DACT's concept is akin to Cinematch on Netflix data as a ground-truth for movie recommendation (and employed in the KDD Cup [5]). DACT contains a collection of trajectories of time-ordered tuples, and provides the following attributes for each tuple.

TID: Unique identifier of the trajectory.

TimeStep: A positive integer standing for the index of the current tuple in the sequence of tuples in the trajectory TID.

TimeStamp: The time of the current tuple, reported in Eastern Daylight Time (EDT).

Speed: The vehicle's speed at the tuple's timestamp, reported in miles per hour (*mph*).



(a) Cohen's Kappa

(b) Jaccard

Figure 4: Inter-Expert Agreement based on (a) Cohen's Kappa and (b) Jaccard.

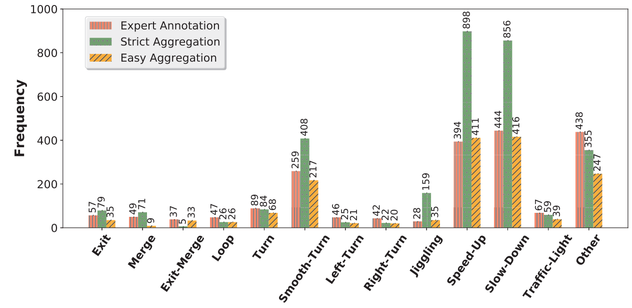


Figure 5: The frequency distribution of segment types as result of different annotation phases.

Table 2: Annotation Facts Based on Different Phases.

Annotation Phase	Num of Segments Specified By Expert(s)	Avg Num of Segments per Trajectory	Total Active Annotation Time	Avg Annotation Time per Trajectory	Avg Time to Specify a Single Segment
Expert Annotation	1,997	20	*17 hours*	*10.3 minutes*	*31 seconds*
Strict Aggregation	2,465	49	*21 hours*	*25.2 minutes*	*31 seconds*
Easy Aggregation	1,372	27	*6.3 hours*	*7.5 minutes*	*15 seconds*

Acceleration: The vehicle's acceleration at the tuple's timestamp reported in meter per second squared (m/s^2).

Heading: The vehicle's direction of moving at the tuple's timestamp, reported in a range $[0, 359]$, where 0 signifies north and 180, south.

HeadingChange: The vehicle's change of heading in comparison with the observed heading in the previous timestep. By definition, the HeadingChange of the first timestep is 0.

Latitude and Longitude: The GPS coordinates of the vehicle's location at the tuple's timestamp.

Annotation: The list of annotations for the tuple (if any). If there is no annotation specified for the tuple, this attribute is set to NULL.

SegmentType: The type(s) of segment for the tuple (if any). The possible values for this attribute are shown in Figure 3. If there is no segment type specified for the tuple, this attribute is set to NULL.

DACT is organized in two CSV files, one for strict-aggregation and the other for easy-aggregation. Columns of the CSV files correspond to the above attributes. DACT is publicly available under the Creative Common license, at the following link: https://goo.gl/XgsxyJ.

5.1 DACT in Practice

DACT is already in use within our research group. In [11], DACT was employed to compare different trajectory segmentation methodologies by performing prediction and sensitivity experiments. There we showed that for the same recall value, how the proposed solution outperformed other baselines (e.g., [2]) by reasonable margin on precision value. Note that without using DACT, there was no other way to quantitatively show the applicability of our proposed segmentation approach in compare to the state-of-the-art solutions.

6 CONCLUSION

In this paper, we introduce a trajectory annotation framework in order to formalize the subjective task of annotating personal car trajectories. In our annotation framework, we address "subjectivity" in decisions, which is the main driver of the task's complexity. For that, we designed a two-step approach where an aggregation phase follows the annotation phase to reduce disagreements between annotators. We also considered an automatic annotator (AutoAnn) which provides baselines to the expert for a better decision making. We illustrated in experiments that our framework boosts inter-annotator agreement from 44% to about 60% (upon Cohen's Kappa metric). We have made the outcome of this research (i.e., the unique dataset of annotated trajectories) publicly available for the spatiotemporal research community.

REFERENCES

[1] New York taxi dataset. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. Accessed: 2017-04-14.

[2] S. Alewijnse, K. Buchin, M. Buchin, A. Kölzsch, H. Kruckenberg, and M. A. Westenberg. A framework for trajectory segmentation by stable criteria. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 351–360. ACM, 2014.

[3] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. Keogh, and P. S. Yu. Global distance-based segmentation of trajectories. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 34–43. ACM, 2006.

[4] B. Aronov, A. Driemel, M. V. Kreveld, M. Löffler, and F. Staals. Segmentation of trajectories on nonmonotone criteria. *ACM Transactions on Algorithms (TALG)*, 12(2):26, 2015.

[5] J. Bennett, S. Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA, 2007.

[6] M. Buchin, A. Driemel, M. van Kreveld, and V. Sacristán. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *Journal of Spatial Information Science*, 2011(3):33–63, 2011.

[7] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[8] M. Haklay and P. Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[9] T. Luo, X. Zheng, G. Xu, K. Fu, and W. Ren. An improved dbscan algorithm to detect stops in individual trajectories. *ISPRS International Journal of Geo-Information*, 6(3):63, 2017.

[10] S. Moosavi, B. Omidvar-Tehrani, R. B. Craig, and R. Ramnath. Annotation of car trajectories based on driving patterns. *arXiv preprint arXiv:1705.05219*, 2017.

[11] S. Moosavi, B. Omidvar-Tehrani, R. B. Craig, R. Ramnath, and A. Nandi. Characterizing driving context from driver behavior. In *Proceedings of the ACM SIGSPATIAL*. ACM, 2017.

[12] S. Moosavi, R. Ramnath, and A. Nandi. Discovery of driving patterns by trajectory segmentation. In *Proceedings of the 3rd ACM SIGSPATIAL PhD Symposium*, page 4. ACM, 2016.

[13] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402, 2013.

[14] C. Panagiotakis, N. Pelekis, I. Kopanakis, E. Ramasso, and Y. Theodoridis. Segmentation and sampling of moving object trajectories based on representativeness. *IEEE Transactions on Knowledge and Data Engineering*, 24(7):1328–1343, 2012.

[15] D. Quercia. Playful cities: Crowdsourcing urban happiness with web games. *Built Environment*, 42(3):430–440, 2016.

[16] D. Quercia and D. Saez. Mining urban deprivation from foursquare: Implicit crowdsourcing of city land use. *IEEE Pervasive Computing*, 13(2):30–36, 2014.

[17] G. Van Brummelen. *Heavenly mathematics: The forgotten art of spherical trigonometry*. Princeton University Press, 2012.

[18] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. Semitri: a framework for semantic annotation of heterogeneous trajectories. In *Proceedings of the 14th international conference on extending database technology*, pages 259–270. ACM, 2011.

[19] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li. *Geolife GPS trajectory dataset - User Guide*, July 2011.