

# Improving Vehicle Localization with Two Low-Cost GPS Receivers



Elnaz Namazi, Rudolf Mester, Chaoru Lu, Markus Metallinos Log,  
and Jingyue Li

**Abstract** A primary concern of Intelligent Traffic Management Systems (ITMSs) is to collect the required traffic data. Vehicle position is one of the most important data types to manage traffic effectively. In this regard, Global Positioning System (GPS) receivers are widely used; however, their estimation accuracy is affected by several parameters, such as signal blockage. Map-matching is one of the most popular approaches to dealing with this challenge. In this study, we investigated the performance of map-matching software and found that it cannot locate the vehicle effectively if the positional data are too noisy. This paper aims to propose a new methodology by integrating cross-GPS validation, interpolation, best-fit, and map-matching techniques to enhance the vehicle localization performance in the presence of GPS signal noise and investigate the methodology with real traffic data from a metropolitan area. Our evaluations indicate that the proposed methodology can significantly improve vehicle self-localization performance.

**Keywords** Vehicle self-localization · GPS receiver · Map-matching

---

E. Namazi (✉) · R. Mester · J. Li (✉)  
Department of Computer Science, Norwegian University of Science and Technology (NTNU),  
Trondheim, Norway  
e-mail: [elnaz.namazi@ntnu.no](mailto:elnaz.namazi@ntnu.no)

J. Li  
e-mail: [jingyue.li@ntnu.no](mailto:jingyue.li@ntnu.no)

R. Mester  
e-mail: [rudolf.mester@ntnu.no](mailto:rudolf.mester@ntnu.no)

C. Lu  
Department of Civil Engineering and Energy Technology, Oslo Metropolitan University  
(OsloMet), Oslo, Norway  
e-mail: [chaorulu@oslomet.no](mailto:chaorulu@oslomet.no)

M. M. Log  
Department of Civil and Environmental Engineering, Norwegian University of Science  
and Technology (NTNU), Trondheim, Norway  
e-mail: [markus.log@ntnu.no](mailto:markus.log@ntnu.no)

# 1 Introduction

Over the past several years, population growth has led to an increase in vehicle numbers, resulting in increased traffic congestion in many cities. As a result, Intelligent Traffic Management Systems (ITMSs) are introduced to manage traffic based on traffic data and make smart decisions. Such data could originate from stationary sensors, such as inductive loop detectors, or from vehicle-mounted sensors, such as Global Positioning System (GPS) receiver, camera, radar, and Light Detection And Ranging (LiDAR).

Vehicle location is one of the most important kind of traffic data. A GPS receiver is a common solution to estimate vehicle location in a GPS coordinate system (also called vehicle self-localization), as most Modern Vehicles (MVs) are equipped with it. However, the accuracy of the data collected via a GPS receiver depends on several parameters, such as hardware accuracy, satellite geometry, signal blockage, and atmospheric conditions [6].

To satisfy vehicle localization requirements and mitigate the estimated location error, three major categories of approaches have been proposed in the literature [3, 5]. One category of approaches uses a standalone reference station, such as Wide Area Augmentation System (WAAS) (e.g., [17]). The second category comprises auxiliary hardware-based approaches (e.g., Inertial Measurement Unit (IMU) [8]). Using technologically advanced sensors to determine vehicle location would boost the estimation accuracy. However, equipping a vehicle with such sensors will also increase the vehicle's cost. The third category uses software, such as map-matching techniques (e.g., [18]). Map-matching is a technique that integrates map information and recorded geolocation data from the vehicle in order to increase the accuracy of the vehicle's location [19]. Although map-matching techniques are widely applied to minimize vehicles' localization error, in this study, we found that map-matching techniques (e.g., QGIS-Plug-in Offline-MapMatching [12, 13]) do not work well if the GPS data collected via a low-cost GPS receiver are too noisy.

Therefore, a much-debated question is how to keep the hardware's and sensors' costs low and the localization performance high. This paper proposes a new methodology by integrating cross-GPS validation, interpolation [4], best-fit [2], and map-matching [12, 13] techniques to localize a vehicle in the presence of GPS signal noise. Our proposed methodology can identify the more accurate GPS receiver dynamically by considering the fixed and known distance between two GPS receivers mounted on the same vehicle. We implemented and evaluated our approach using real traffic data from a metropolitan area in Chengdu, China. The results show that our proposed approach can enhance vehicle self-localization performance.

The paper is organized as follows. Section 2 gives a brief overview of related work. Section 3 explains our proposed research design. Section 4 presents our proposed research approach. Section 5 describes our evaluation of the methodology. Section 6 presents the discussion. The last section concludes and proposes future studies.

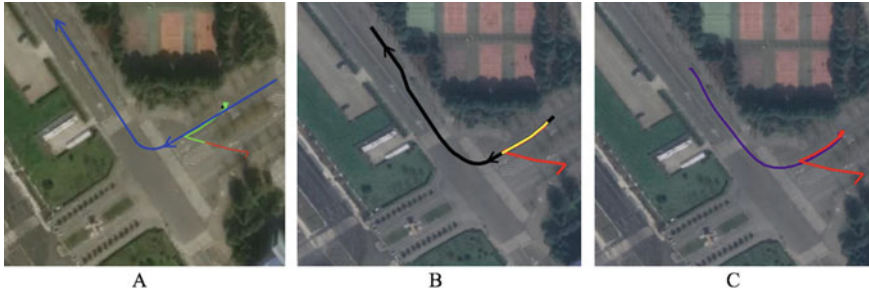
## 2 Related Work

Vehicle localization based on GPS receivers is a key component in managing traffic safely and effectively. However, it can be imprecise, causing operational difficulties. Many approaches have been proposed to process imprecise data from GPS receivers to acquire accurate vehicle localization. For instance, Islam et al. [5] enhanced GPS accuracy by considering a vehicle's movement direction, velocity averaging, and the distance between waypoints using coordinate data. Their experiment used a vehicle-mounted Garmin GPS 19xHVS receiver. In order to examine the accuracy, they plotted the data on Google Maps. The proposed approach achieved improvement of 4–10 m [5]. Acosta et al. [11] proposed an approach based on a Kalman, fuzzy logic, and information selection. In the experimental step, they used three Garmin 18X GPS receivers that were connected to two notebooks. The proposed approach in [11] smoothed the measurement error and mitigated the error that fluctuates in time. Tang et al., in [15], proposed an adaptive map-matching algorithm based on a hierarchical fuzzy system. The experimental results showed that the proposed algorithm in [15] was able to increase the matching accuracy and to outperform the traditional algorithms based on only geometric or topological information of network. Lecce et al. [1] used generalized regression neural networks to increase the GPS position accuracy by correcting the receiver's position. The idea was to use an analytical description of the time series to improve the position accuracy. They proposed an approach based on removing the GPS positioning error by training a neural network to mitigate the periodic components of the GPS positioning error. In the experimental step, they used GPS receiver BU-353. The mean improvement in the accuracy of the GPS position of the proposed approach was 25% [1].

## 3 Research Design

Our recent studies [9, 10] proposed new methodologies that use the ego-vehicle as a mobile sensor, estimating the traffic data for surrounding vehicles in order to share them with an ITMS. Vehicle localization (i.e., ego-vehicle and target vehicle localization) plays an important role in managing the traffic. Our studies revealed that the image-based target vehicle localization accuracy is tightly connected to the localization accuracy of the ego-vehicle.

This paper aims to enhance the ego-vehicle localization performance by using two low-cost GPS receivers. Each vehicle was equipped with two monocular cameras and built-in GPS receivers. One camera was mounted on the front window of the vehicle, and another was mounted on the rear window. These two cameras were located at a known distance from each other on the vehicle, helping us to validate the GPS receiver accuracy, as well as collecting footage from both sides of the vehicle, which was needed for further image processing-based studies. All cameras used were of the type GoPro Hero 7. The monocular camera was a low-cost sensor that can be mounted on most ego-vehicles, making our approach generalizable. In addition to collecting video footage, the chosen camera enabled GPS data collection, as it included a built-



**Fig. 1** Problem formulation. A) Vehicle locations collected via a front-mounted GPS receiver on the vehicle (green-red polyline), compared with the vehicle’s movement scenario (blue polyline). B) Map-matching output (yellow polyline) related to the noisy front GPS receiver (red polyline) by considering the true trajectory of the vehicle (black polyline). C) Vehicle locations obtained via two GPS receivers on the same vehicle (front GPS receiver: red polyline; rear GPS receiver: purple polyline)

in GPS receiver (hereafter, we call this camera as a GPS receiver since it embeds a GPS receiver).

To begin this research, we first analyzed the accuracy of the collected GPS data obtained via the GPS receiver mounted on the front window glass by plotting the data on a map (the data collection process is described in detail in Sect. 5.1). Figure 1 shows one example of the studied scenarios in which the ego-vehicle turns right at an intersection. In Fig. 1-A, the blue arrow shows the vehicle’s movement scenario. The polyline, which is a combination of green and red colors, represents the vehicle’s location based on the front GPS receiver mounted on the vehicle. The color of the polyline represents the vehicle’s speed. This polyline and its colors are plotted automatically by using “Telemetry Extractor for GoPro” [16].

Our first attempt was to use map-matching software to address the GPS receiver noise issue to obtain the precise vehicle location. We used the QGIS-Plug-in Offline-MapMatching [12, 13], which is one of the widely used approaches for minimizing the GPS receiver error. Figure 1-B shows our findings after applying map-matching to the same scenario shown in Fig. 1-A. In Fig. 1-B, the black polyline is the true trajectory of the vehicle on the road. The red polyline represents the positions collected via the front GPS receiver mounted on the vehicle (part of this red polyline is covered by the yellow polyline), and the yellow polyline represents the map-matched positions of the noisy GPS receiver. From this figure, it is clear that the QGIS-Plug-in Offline-MapMatching [12, 13] is not able to identify and map-match the entire trajectory accurately if the vehicle localization error is too high.

We then analyzed data from another GPS receiver on the same vehicle in the scenario shown in Fig. 1-A and Fig. 1-B. Figure 1-C shows the results. In Fig. 1-C, the red polyline is the vehicle’s position based on the GPS receiver mounted on the front window. The purple polyline shows the vehicle’s position based on the GPS receiver mounted on the rear window on the same vehicle. As Fig. 1-C shows, the localization error of the front-mounted GPS receiver is much higher than that of the rear-mounted GPS receiver in this scenario.

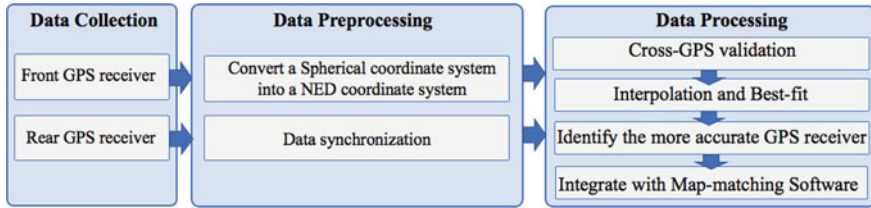


Fig. 2 Our proposed research approach.

## 4 Research Approach

Figure 2 illustrates our proposed research approach, comprising data collection, data preprocessing, and data processing.

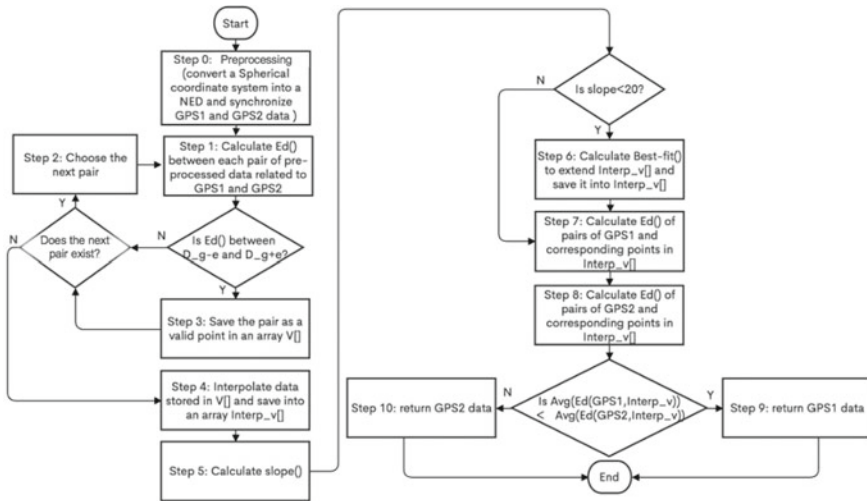
### 4.1 Data Preprocessing

As previously stated, positional data of our study were collected using two GPS receivers mounted on the ego-vehicle. Before applying our approach, the data were preprocessed. In this step, firstly, we need to convert a Spherical coordinate system [14] into a local North-East-Down (NED) coordinate system [7] on the earth’s surface. The conversion is both practical and justified, since we are studying a small, demarcated area on the earth’s surface. Secondly, since the two mounted GPS receivers on the vehicle are independent and the data collection was not started concurrently, we need to synchronize the GPS receivers in the time domain.

### 4.2 Data Processing

In this step, first, we need to analyze the accuracy of the two mounted GPS receivers on the same vehicle. To detect whether the GPS signals are accurate, we calculated the vector distance of the estimated positions obtained via the two GPS receivers at equal timestamps, as the two GPS receivers were mounted with a fixed and known distance from each other (in our study, we assumed this fixed distance  $D_g$  is 3 m, because we used family vehicles) on the same vehicle. If it is found that the vector distances are different from this fixed distance (with an error threshold  $e = \pm 2$  m), we can conclude that at least one of the GPS receivers is inaccurate, which means we need to identify the more accurate GPS receiver.

To identify the more accurate GPS receiver, we developed a new algorithm based on cross-validation, interpolation [4], and best-fit [2] techniques, as presented in Fig. 3. Cross-validation found the positions in the trajectory where both GPS receivers



**Fig. 3** Flowchart of our proposed algorithm.

were almost in agreement on the vehicle's position (i.e., the position difference obtained by two GPS receivers was between  $D_g - e$  and  $D_g + e$ ). It did so based on the Euclidean distance (Ed) between each pair of preprocessed positions obtained by the front and rear GPS receivers per each timestamp. In this study, we assumed that the error of the GPS receiver was random error, which means that the GPS receivers can obtain accurate locations in most time (hereafter, the accurate locations are called valid points). Due to possible perturbations [6], GPS receivers can sometimes provide noisy locations. Our idea is to identify the valid points and then use interpolation technique [4] to calculate the possible locations when the GPS error is identified. In addition, for the straight vehicle movements, which were determined based on the vehicle's movement slope (we assumed the movement with a slope less than  $20^\circ$  as a straight movement, otherwise as a turn), the best-fit technique [2] was used to generate more positions in the whole trajectory based on the interpolated positions. To identify the more accurate GPS receiver, we then calculated the average Euclidean distance between the positions calculated through interpolation and best-fit and the positions collected by each GPS receiver. The GPS receiver with the smallest average Euclidean distance was identified as the more accurate one.

Although we can identify that one GPS receiver is more accurate than the other, the more accurate one may also be noisy. Finally, we inserted the data from the identified more accurate GPS receiver into a map-matching algorithm, using it to further amend the noisy GPS signal. We investigated the effectiveness of several existing map-matching software applications and identified the one that was most compatible with our data. We found that the QGIS-Plug-in Offline-MapMatching [12, 13] was a suitable and effective tool for map-matching in our research context.

## 5 Evaluation

### 5.1 Data Collection

To evaluate our proposed approach, experiments were run using several case studies with real traffic data. We used three equipped vehicles (each vehicle were equipped with two GPS receivers, as described in Sect. 3) driven in the metropolitan region of Chengdu, China. In order to provide good data coverage and generalizability, eight different scenarios were defined, comprising both straight-street and intersection movements. In total, 24 trajectories were considered. There were many tall buildings around the studied area, which may interfere with GPS signal accuracy and cause GPS data inaccuracies. As the ground truths related to vehicle movements in this study were not available from the GPS receiver data, we extracted them manually by visually observing forward-facing video footage and identifying the ground-truth vehicle movements using Google Earth Pro [20].

### 5.2 Evaluation of the Results

As we observed in Fig. 1, if the GPS signal was too noisy, the QGIS-Plug-in Offline-MapMatching [12, 13] was able to minimize the localization inaccuracy of only a segment of the trajectory. We used the Cartesian length of the trajectory to evaluate the performance of our proposed self-localization approach. Table 1 summarizes our findings. This table included eight scenarios (S1–S8) and three equipped vehicles (V1–V3). The “ground truth” column shows the Cartesian length of a vehicle’s movement, and the “avg. dis.” columns represents the average distance between the vehicle positions collected via each GPS receiver and the ground truth. The GPS receiver with a smaller average distance was labeled as a more accurate GPS receiver. To assess our proposed methodology, we first calculated the Cartesian length of the vehicle trajectory by using only map-matching on data of both front and rear GPS receivers. Our findings are presented in the “front GPS” and “rear GPS” sub-columns of the “map-matching based Cartesian length” column. We then calculated the Cartesian length of the vehicle trajectory, after applying our proposed methodology and identifying the accurate GPS receiver. The results are presented in the “accurate GPS” (results of steps 9 and 10 in Fig. 3) and “Cartesian length” sub-columns of the “our proposed approach” column. In addition, we compared the deviation from the ground truth by using only map-matching on the collected data and using our proposed approach. The results are shown in the sub-columns of the “deviation comparison” column.

To explain the information presented in Table 1 in depth, we use scenario S8 and vehicle V3 as an example. In this scenario, the Cartesian lengths of the map-matched positions obtained via both GPS receivers are almost the same (front GPS: 150 m; rear GPS: 148 m). This shows that applying map-matching software would be enough to correct such small errors satisfactorily. However, this table shows that when the

**Table 1** Case study evaluation.

S#	V#	Ground truth (m)	Avg. Dis. (m)		Map-matching-based Cartesian Length (m)		Our proposed approach		Deviation comparison (m)		
			Front GPS	Rear GPS	Front GPS	Rear GPS	Accurate GPS	Cartesian Length (m)	Front GPS	Rear GPS	Our proposed approach
S1	V1	532	12.009	4.935	490	532	Rear	532	-42	0	0
	V2	514	2.044	10.746	513	514	Front	513	-1	0	-1
	V3	441	2.324	4.415	441	437	Front	441	0	-4	0
S2	V1	179	1.457	6.058	179	178	Rear	178	0	-1	-1
	V2	191	4.358	3.385	191	177	Front	191	0	-14	0
	V3	147	1.669	2.19	147	145	Front	147	0	-2	0
S3	V1	191	1.955	1.774	191	155	Rear	155	0	-36	-36
	V2	189	1.552	4.612	189	184	Front	189	0	-5	0
	V3	159	3.608	13.860	159	150	Front	159	0	-9	0
S4	V1	159	4.241	0.665	156	159	Rear	159	-3	0	0
	V2	163	6.044	1.84	163	162	Front	163	0	-1	0
	V3	188	1.388	2.264	188	188	Front	188	0	0	0
S5	V1	174	5.170	1.798	174	162	Rear	162	0	-12	-12
	V2	188	3.126	4.900	188	188	Front	188	0	0	0
	V3	118	1.450	2.385	118	118	Front	118	0	0	0
S6	V1	124	3.752	6.913	124	117	Rear	117	0	-7	-7
	V2	194	1.333	7.131	186	194	Front	186	-8	0	-8
	V3	-	-	-	-	-	-	-	-	-	-
S7	V1	106	1.834	4.460	106	106	Rear	106	0	0	0
	V2	142	7.660	4.803	141	142	Front	141	-1	0	-1
	V3	-	-	-	-	-	-	-	-	-	-
S8	V1	<b>109</b>	<b>3.515</b>	<b>1.402</b>	<b>27</b>	<b>109</b>	<b>Rear</b>	<b>109</b>	<b>-82</b>	<b>0</b>	<b>0</b>
	V2	107	1.493	2.983	103	107	Front	103	-4	0	-4
	V3	150	1.627	2.939	150	148	Front	150	0	-2	0

GPS error is high, applying only map-matching may not be effective, which is the main focus of this study. For instance, in scenario S8 and vehicle V1, the Cartesian length by applying map-matching associated with the front GPS receiver is 27 m, while it is 109 m for the rear GPS receiver. It means by using only one GPS receiver (i.e., front GPS receiver), map-matching is only effective for a small segment of the trajectory (i.e., 27 m). The performance could be increased if we consider another GPS receiver. It confirms that identifying the more accurate GPS receiver is vital, which is the rear GPS receiver in this case. After identifying the more accurate GPS receiver and using its collected data to feed into the map-matching software, our proposed approach increased the self-localization performance which is measured using the Cartesian length of the output to 109 m. Therefore, the S8 and V1 case showed that our approach is effective in the presence of extreme GPS signal noise. In Table 1, using our approach to choose a more accurate GPS receiver first and then use map-matching does not always give less deviation than using front or rear GPS receiver randomly. The reason is that we chose to use GPS2 data in step 10 in Fig. 3 when data from both GPS receivers were acceptable. The GPS2 data may not be



better than GPS1 data in some cases, although both data from both GPS receivers are acceptable. In this table, for vehicle V3 in scenarios S6 and S7, information are not provided, as the rear GPS receiver did not record during the whole scenario. The reason for this could be that the battery died or that the memory card became full.

## 6 Discussion

Previous studies have noted the importance of identifying and mitigating the measurement error of GPS receivers. This paper developed a new algorithm to identify the more accurate GPS receiver if there are multiple, possibly noisy, GPS receivers installed on the same vehicle, based on cross-validation, interpolation [4], and best-fit [2] techniques. Compared to the approach relying on expensive GPS receivers, our approach provides a low-cost solution to identify a vehicle's location precisely. Compared to the approach that relies solely on map-matching, our strategy of detecting GPS inaccuracy and prioritizing the data from the more accurate GPS receiver helped enhance the performance of the map-matching software, when the GPS signal is too noisy. One of the limitations in this study is that the cross-validation step is limited to address random GPS receiver error, which means that our approach relies on the existence of valid points, as explained in Sect. 4.2, which are collected by both GPS receivers on the same vehicle. If the localization error of one GPS receiver is too high and there are no overlapping points between GPS receivers in the studied trajectories, cross-validation is infeasible. Moreover, the current approach and evaluation are based on postprocessing and the QGIS-Plug-in Offline-MapMatching [12, 13]. By analyzing only a small segment of a trajectory each time and using real-time map-matching software, it would be possible to turn the solution to be more real-time.

## 7 Conclusion and Future Work

In this study, our research context is defined as mounting two low-cost and possibly imprecise GPS receivers on the same vehicle at a fixed and known distance from each other to accurately identify the position of the vehicle based on cross-validation, interpolation, and best-fit techniques while the vehicle is moving. We developed a new algorithm to identify the more accurate GPS receiver in the presence of noise and fed the GPS data from the more accurate GPS receiver into map-matching software. The proposed approach minimized the measurement error of the low-cost GPS receiver and was able to enhance the vehicle localization performance. Since the study was limited to vehicle movements through intersections and along straight streets with limited scenarios, more studies are needed to be able to generalize our approach by considering various vehicle movements, driving speeds, and weather conditions.

## References

1. V. Di Lecce, A. Amato, V. Piuri, Neural technologies for increasing the GPS position accuracy, in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications 2008* (IEEE, 2008), pp. 4–8
2. How do I construct a straight line through data points? Best-fit lines, <https://serc.carleton.edu/mathyouneed/graphing/bestfit.html>
3. J.Y. Huang, C.H. Tsai, Improve GPS positioning accuracy with context awareness, in *First IEEE International Conference on Ubi-Media Computing 2008* (IEEE, 2008), pp. 94–99
4. Interpolation Methods, <https://www.mathworks.com/help/curvefit/interpolation-methods.html>
5. M. Islam, J.M. Kim, An effective approach to improving low-cost GPS positioning accuracy in real-time navigation. *Sci. World J.* **2014**, 1–8 (2014)
6. J.K. Laphong, H. Dihingia, S. Borah, Advanced road traffic management system. *IJISSET* **4**, 168–169 (2019)
7. Local tangent plane coordinates, [https://en.wikipedia.org/wiki/Local\\_tangent\\_plane\\_coordinates](https://en.wikipedia.org/wiki/Local_tangent_plane_coordinates)
8. H. McCall, C.F. Lin, American GNC Corp, assignee. Micro integrated global positioning system/inertial measurement unit system. United States patent application US (2002)
9. E. Namazi, J. Li, R. Mester, C. Lu, Identifying and counting vehicles in multiple lanes by using a low-cost vehicle-mounted sensor for intelligent traffic management systems, in *HAIIS 2020*, ed. by E.A. de la Cal, J.R. Villar Flecha, H. Quintián, E. Corchado. LNCS (LNAI), vol. 12344 (Springer, Cham, 2020), pp. 598–611. [https://doi.org/10.1007/978-3-030-61705-9\\_49](https://doi.org/10.1007/978-3-030-61705-9_49)
10. E. Namazi, R.N. Holthe-Berg, C.S. Lofsberg, J. Li, Using vehicle-mounted camera to collect information for managing mixed traffic, in *15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) 2019* (IEEE, 2019), pp. 222–230
11. N. Acosta, J. Tolozza, Techniques to improve the GPS precision. *Int. J. Adv. Comput. Sci. Appl.* **3**(8), 125–130 (2012)
12. Offline-MapMatching, <https://github.com/jagodki/Offline-MapMatching>
13. QGIS, <https://qgis.org/en/site/>
14. Spherical coordinate system, [https://en.wikipedia.org/wiki/Spherical\\_coordinate\\_system](https://en.wikipedia.org/wiki/Spherical_coordinate_system)
15. J. Tang, S. Zhang, Y. Zou, F. Liu, An adaptive map-matching algorithm based on hierarchical fuzzy system from vehicular GPS data. *Plos One* **12**(12), e0188796 (2017)
16. Telemetry Extractor for GoPro, <https://goprotelemetryextractor.com/free/>
17. T.H. Witte, A.M. Wilson, Accuracy of WAAS-enabled GPS for the determination of position and speed over ground. *J. Biomech.* **38**(8), 1717–1722 (2005)
18. Z. Wu, J. Xie, Y. Wang, Y.M. Nie, Map matching based on multi-layer road index. *Transp. Res. Part C: Emerg. Technol.* **118**, 102651 (2020)
19. L.X.Q. Liu, M.L.Z. Liu, Map matching algorithm and its application, in *Proceedings on Intelligent Systems and Knowledge Engineering (ISKE2007)* (2007)
20. Google Earth, <https://www.google.com/earth/download/gep/agree.html?hl=en-GB>