

A Machine Learning Approach to Improve the Accuracy of GPS-Based Map-Matching Algorithms

Mahdi Hashemi, Hassan A. Karimi

Geoinformatics Laboratory, School of Information Sciences
University of Pittsburgh
Pittsburgh, PA
m.hashemi1987@gmail.com, hkarimi@pitt.edu

Abstract—Advanced map-matching algorithms use location and heading of GPS points along with geometrical and topological features of digital road networks to find the road segment on which the vehicle is moving. However, GPS errors sometimes impede map-matching algorithms in finding the correct segment, especially in dense and complicated parts of the network, such as near intersections with acute angles or on close parallel roads. In this paper an artificial neural network (ANN) approach is explored to improve the segment identification accuracy of map-matching algorithms. The proposed ANN is continuously trained by using the horizontal shift imposed on GPS points and once it is trained, it will be used to correct raw GPS points before inputting them into the map-matching algorithm. Integrating the proposed ANN enabled an existing map-matching algorithm to find the correct segments for some of the GPS points where the original map-matching algorithm had failed to do so.

Keywords—Machine learning; Map-matching algorithm; Road network; GPS error; Spatio-temporal pattern.

I. INTRODUCTION

GPS accuracy is degraded by atmospheric conditions, multipath (where the signal from satellites is reflected by objects such as buildings, dense vegetation, tunnels, and large vehicles), signal attenuation, receiver noise (including received signal power and interference power), receiver resolution, and low speed [1; 2; 3]. The GPS positioning accuracy ranges from 12.7 to 24.1 m in 95% of the time during normal operation [2]. Still, this range is subject to change because the receiver's performance is highly uncertain under different environmental conditions. With a HDOP (horizontal delusion of precision) of 1.8 and a very conservative 12.3 m 1-sigma UERE (user equivalent range error or total pseudorange inaccuracy) value, the horizontal error may reach up to 38.2 m in 95% of the time at worst [2]. However, hills, dense trees and tall buildings surrounding the roads, which may block GPS signals, are not taken into account by [2]. Karimi et al. [4] estimated that errors stemmed from ionosphere, troposphere, measurement noise, ephemeris data, clock drift, and multipath could reach up to 100 m, 30 m, 5 m, 5 m, 1.5 m, and 25 m, respectively. Other researchers have mentioned different horizontal accuracies for GPS points such as 13 m at 95% confidence level [5; 6; 7], 20 m at 95% confidence level [8], 10 m [9],

15 m [10], 30 m [11], 50 m [12], and 80 m [13]. The reason for such a wide range in GPS error value is the difference in the environment and time of measurement (satellites have different constellations at different times). Positioning in a narrow street surrounded by tall buildings and trees would result in the worst GPS accuracy.

Research in map-matching algorithms, which are at the heart of any navigation systems, has been focused on how to match newly obtained GPS data with road networks [14; 15; 16; 17; 18]. However, due to GPS errors, as explained above, even advanced map-matching algorithms sometimes are unable to find the correct road segment [12; 19]. As a less costly alternative approach to hardware-based approaches, such as differential GPS (DGPS), an artificial neural network (ANN) is explored in this work to reduce GPS horizontal errors before GPS points are inputted into the map-matching algorithm. The proposed ANN is trained by the results of the map-matching algorithm and is used to improve the horizontal accuracy of the proceeding GPS points before they are inputted into the map-matching algorithm.

II. BACKGROUND

Much research has been focused on improving poor positional accuracy of GPS points for map-matching algorithms. DGPS [20; 21; 22; 23; 3], inertial measurement unit [24], dead reckoning (DR) [20; 1; 19; 21; 22; 12; 25], compass, accelerometer and camera [6; 26] are among the approaches for that purpose. Of these approaches, DGPS has been implemented widely where one GPS receiver with known precise position (base) is in charge of monitoring the GPS errors and transmitting corrections of errors to other nearby receivers [3]. Therefore, to implement real-time DGPS in vehicle navigation, pseudorange corrections must be continuously transmitted to the vehicle's positioning system through terrestrial radio waves. To bypass the need for a base station, Taylor et al. [3] proposed virtual DGPS where the corrections applied on previous GPS points during map matching are used to calculate pseudorange error of each satellite; pseudorange errors calculated at the previous step are used to modify pseudoranges of those satellites at next GPS positions. They also used the height information provided by digital terrain model (DTM) to reduce the minimum number of required satellites. A least squares model was used to combine all this information and

adjust the position of the vehicle. One of the latest approaches for improving GPS positional accuracy in car navigation systems in areas with poor GPS signal availability or high multipath effect (high HDOP) is to use nearby cars with a good GPS positioning accuracy (low HDOP) as a base station in a DGPS configuration [23]. However, this approach requires a communication link between the nearby cars. Most current approaches require installation of additional hardware alongside the roads and/or on cars. Besides, DR aims to fill the gaps in places where GPS signals are blocked and does not necessarily improve the horizontal accuracy [12; 19; 27]. Our work in this paper explores a software-based approach to increase the correct segment identification percentage of map-matching algorithms by improving the accuracy of raw GPS points without installing additional hardware or considering external data.

III. METHODOLOGY

An ANN is proposed in this work, as a software-based approach, for reducing the horizontal error in GPS points before map matching. Map-matched points are used to train the ANN and the trained ANN is used to horizontally shift the next GPS points before inputting them into the map-matching algorithm. Therefore, no external data or devices are required for training the ANN and the only cost is the additional processing time. This horizontal shift has two components: across-track and along-track. The across-track component increases the chances of identifying the correct segment and the along-track component improves the along-track accuracy of the projected GPS point. In other words, reducing the horizontal error in GPS points not only facilitates the work of map-matching algorithms in identifying the correct segment, but also improves the along-track accuracy of the projected points. The work in this paper is based on the following assumptions:

- The map-matching process reduces the horizontal error in GPS points in most cases.
- Horizontal errors in GPS points are spatially and temporally autocorrelated.

The first assumption is valid based on the known fact that map-matching algorithms are expected to identify correct segments in most cases [16]. A review of latest map-matching algorithms reveals that most algorithms identify correct segments in more than 90% of cases [14; 7; 28; 29; 30; 31]. When the road network database is of high accuracy (more accurate than GPS points), projecting a GPS point on the correct segment reduces its horizontal error. Besides, in map matching, the GPS error relative to the road network is more important than its absolute error (its error relative to its true location).

The second assumption can be validated by examining GPS signal behavior under actual field conditions [10]. Jagadeesh [10] proved that the error difference between consecutive GPS points is much smaller than the absolute GPS error, regardless of GPS receiver type. The error difference reported was under 2 m for over 95% of the cases. A steep change in error between consecutive GPS points is rare and usually occurs because the GPS receiver suddenly fails to receive signals from the required minimum number of satellites or when it regains them. Socharoentum and Karimi [23] and Mosavi [32] also confirmed that GPS error sources (such as atmospheric effects and delays) are correlated over space and time. Therefore, the horizontal error in a GPS point tends to be similar to the errors in its immediate GPS points. Based on this observation, the proposed ANN in this work must be continually trained because training the ANN in one location and time epoch and then using it in a very far location or time epoch would not improve the horizontal accuracy of GPS points.

To train and use the ANN, two sets of data are required. One set is input data, including speed, HDOP and a constant value in this work. However, speed is negated (e.g., if the speed is 15 m/s, it is considered as -15 when it is inputted to the ANN) because its effect on the output is reverse, i.e., the faster the vehicle's speed, the less the GPS horizontal error is [1; 25; 14]. The second set is output data which is the horizontal shift in the GPS point as a vector, represented by length and direction, as shown in Fig. 1.

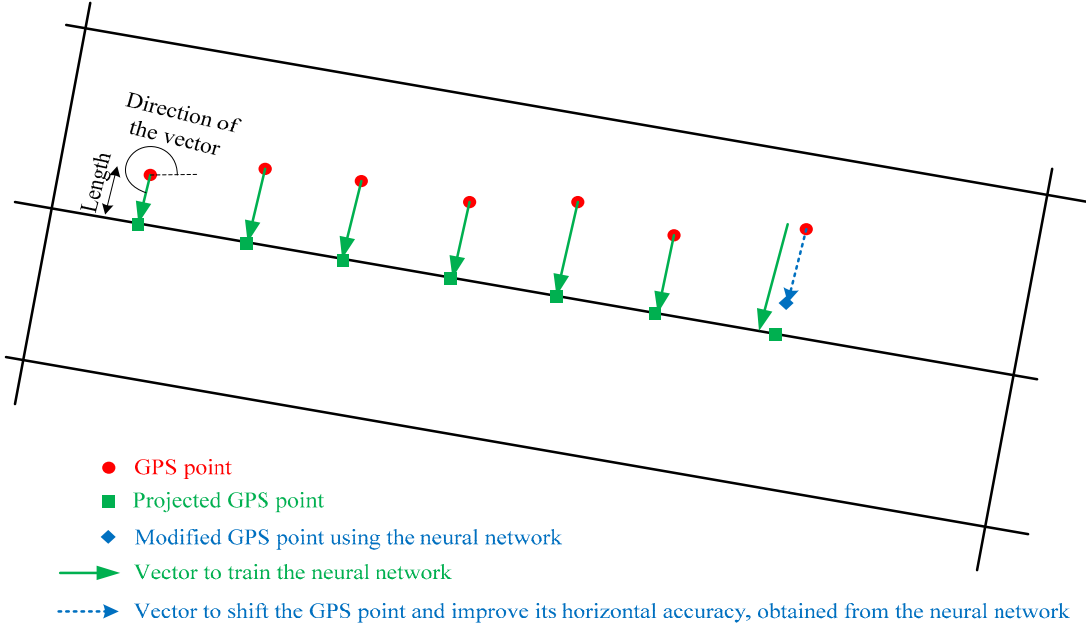


Figure 1. The shift vector connecting the raw GPS point to the projected GPS point is used to train the ANN.

Fig. 2 shows how the proposed ANN should be integrated with a map-matching algorithm to be trained and used to improve the horizontal accuracy of the GPS points, which in turn improves the possibility of identifying correct segments. Two important points are noteworthy here. First, the GPS point with the modified coordinates, which is the output of the ANN, is used in the map-matching algorithm not the raw GPS point. Second, the vector connecting the raw GPS point to the projected GPS point is used to train the ANN not the vector connecting the modified GPS point to the projected GPS point.

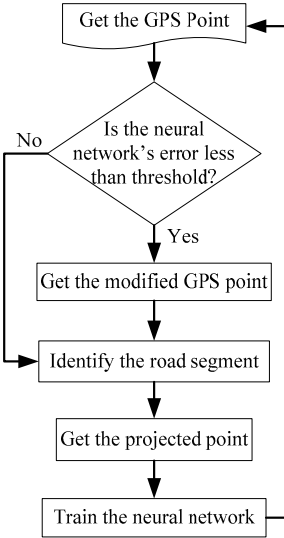


Figure 2. The proposed integration methodology.

The shift vector to modify the GPS point has two components: length and direction. Two ANNs are required, one for estimating the length of the horizontal error and another for estimating its direction. A two-layer feed-forward ANN is used for the length component and another for the direction component. The architecture of the ANN is explained in the next section. Training the two ANNs starts with the first GPS point and continues for all subsequent points. However, the ANNs are used to estimate the length and direction of the horizontal error for next GPS points only when the ANNs' error is less than a threshold. An ANN's error is the difference between its actual output and desired output. The error of an ANN is calculated and updated each time it is trained using an input-output dataset. The input-output dataset is first used to measure the ANN's error representing how close the actual output is to the desired value and then is used to train the ANN.

A. A two-layer feed-forward ANN with back-propagation training rule

Equations 1 and 2 are used to train the ANN. Propagation rules in these equations determine the value of each unit in the hidden and output layers, respectively [33; 34]. The calculated value for each unit is simply the weighted sum of the separate outputs from each of the connected units plus an offset term (θ). Equations 3 and 4 determine the difference between the desired and actual values of each unit [35]. These differences are used in back-propagation rules in Equations 5 and 6 to adjust the weights [35]. Fig. 3 shows what each variable in these equations represents.

$$\begin{aligned} H_h &= \left(\sum_i w_{ih} \times I_i \right) + \theta_h & (1) \\ O_o &= \left(\sum_h w_{ho} \times H_h \right) + \theta_o & (2) \\ \delta_o &= \text{desired output} - \text{actual output} & (3) \\ \delta_h &= \sum_o \delta_o \times w_{ho} & (4) \\ \Delta w_{ho} &= \gamma \times \delta_o \times H_h & (5) \\ \Delta w_{ih} &= \gamma \times \delta_h \times I_i & (6) \end{aligned}$$

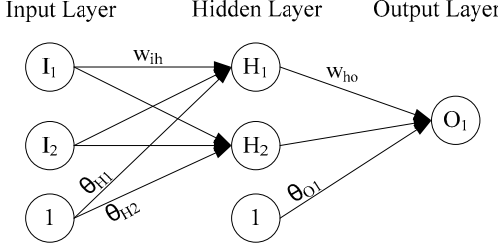


Figure 3. A two-layer feed-forward ANN where the offset is implemented as a weight from a unit with value 1.

A large number of hidden units increases the risk that the function, represented by the ANN, may be more complex than the desired function [33]. Since the training process is online (the training and testing datasets become available in real time) and the ANN functionality and error is dependent on both location and time, the only way to tune the size of the ANN is by trying different sizes.

A large training rate (γ) leads to oscillation [33; 34], i.e., the function represented by the ANN jumps back and forth over the desired function instead of getting closer to it. A small training rate is more conservative and prevents the oscillation but slows down the training process. The training rate must be as large as possible as long as it does not lead to oscillation [33]. Again since the training process is online and the functionality of the ANN changes over space and time, a constant training rate is more reasonable than a dynamic one. However, different values are experimented for this constant and discussed in the Results section.

IV. RESULTS

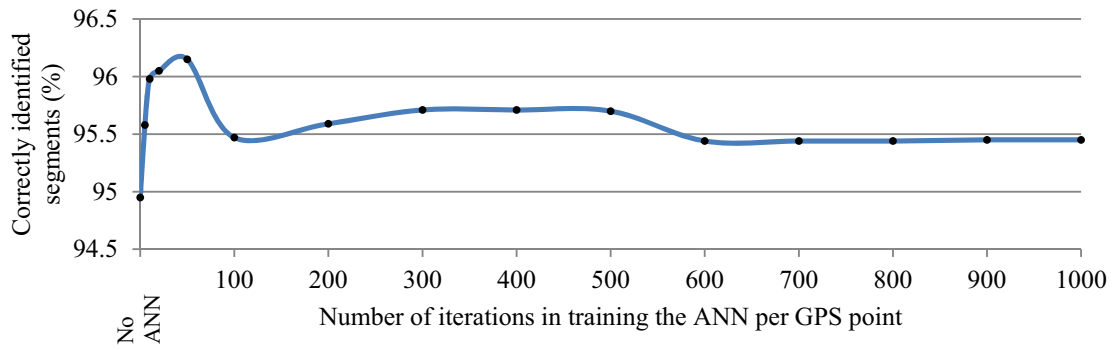
The proposed ANN is integrated with the real-time map-matching algorithm proposed by Hashemi and Karimi [14] based on Fig. 2. This algorithm [14] identifies the best road segment based on three criteria: closeness, heading difference, and direction difference whenever an intersection is detected by the algorithm. Otherwise, the GPS point is projected on the same segment as before. The rationale for choosing this map-matching algorithm for this work is that it is one of the latest map-matching algorithms with good accuracy and time performance tested in dense urban networks. However, the algorithm is not able to identify the correct segment when the GPS error is large and

the point is located around intersections with acute angles or between close parallel roads [14]. These are situations where position, heading, and direction of GPS points lead the map-matching algorithm to the wrong segment. Integrating the proposed ANN with the map-matching algorithm is expected to enable the algorithm to identify the correct segment for some of those challenging situations where the original map-matching algorithm was not able to. Five different GPS trajectories in Chicago obtained from OpenStreetMap (OSM; www.openstreetmap.org) containing a total of 13204 GPS points are used to compare the performance of the map-matching algorithm before and after applying the ANN. The digital road network of Chicago with an accuracy of 5 m was obtained from NAVTEQ (www.navteq.com) for this experiment.

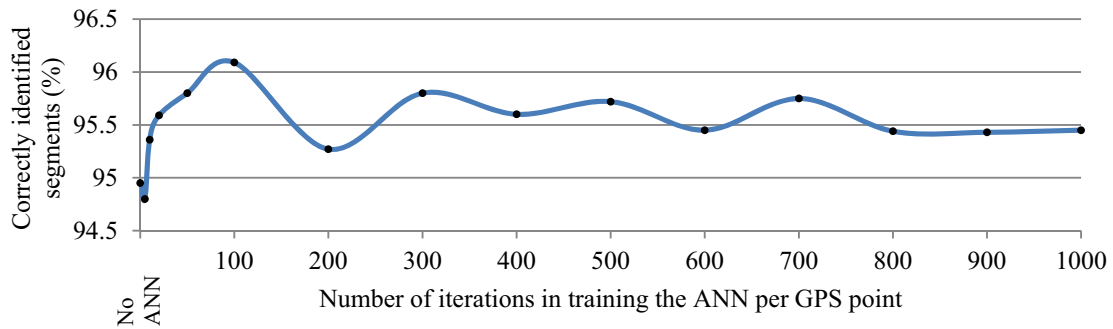
The acceptable error in the ANN for calculating the length of the shift vector is set at 3 m and the acceptable error in the ANN for calculating the direction of the vector is set at 30° . Whenever the errors in the ANNs exceed these thresholds, they will not be used to modify the subsequent GPS points (see Section 3 for how the ANN's error is calculated). If the traveled distance from the previous GPS point exceeds 50 m, the ANN's error is set to a large number because, as explained in Section 3, a gap in the training dataset invalidates the ANN's settings. Shifting GPS points by the ANN is limited to 5 m to prevent large moves in GPS points which may result in a sequence of wrong segments and wrong training datasets. The aforementioned thresholds are chosen based on different observations on the performance of the algorithm, though the sensitivity of the results with these thresholds is very low. The effects of four following items in the ANN (which results in 168 different settings for the ANN) on the performance of the map-matching algorithm are investigated:

- Inputs to the ANN: (a) speed and HDOP, (b) speed, and (c) a constant value
- Number of hidden units: 10 and 20
- Training rate: 0.01 and 0.001
- Number of iterations in training the ANN per GPS point: 5, 10, 20, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000

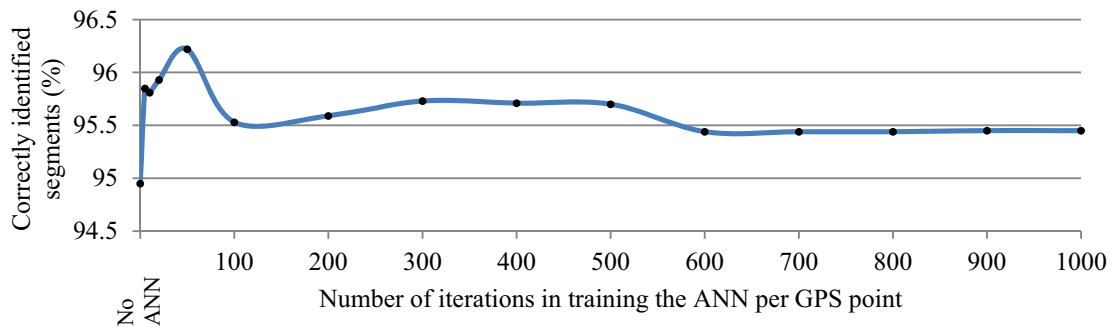
The twelve diagrams in Fig. 4 show the results of the ANN integrated with the map-matching algorithm for different inputs, different numbers of hidden units, and different training rates. There are 14 points in each diagram showing the percentage of correctly identified segments for different numbers of iterations in training the ANN per GPS point. The beginning of each diagram shows the percentage of correctly identified segments before integrating the ANN in the map-matching algorithm.



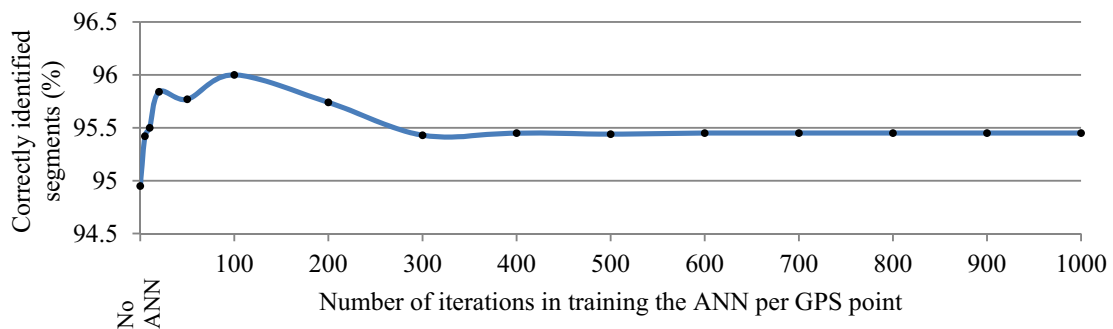
a) Inputs to the ANN: A constant value; Number of hidden units: 10; Training rate: 0.01; Average processing time for one GPS point: 4.31687 millisecond.



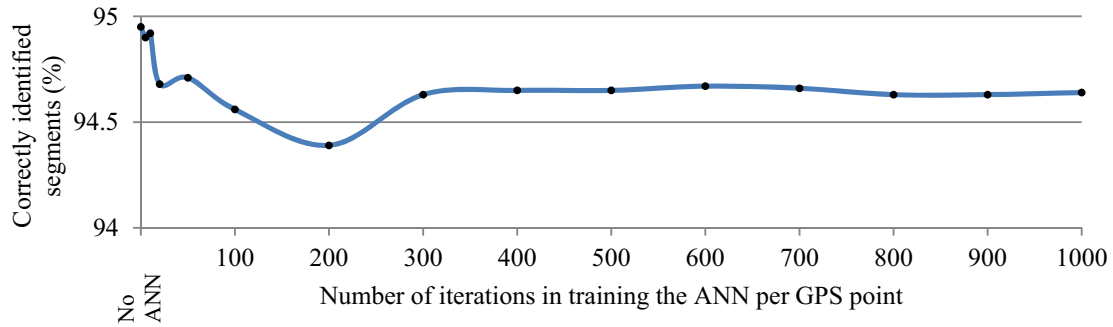
b) Inputs to the ANN: A constant value; Number of hidden units: 10; Training rate: 0.001; Average processing time for one GPS point: 4.24114 millisecond.



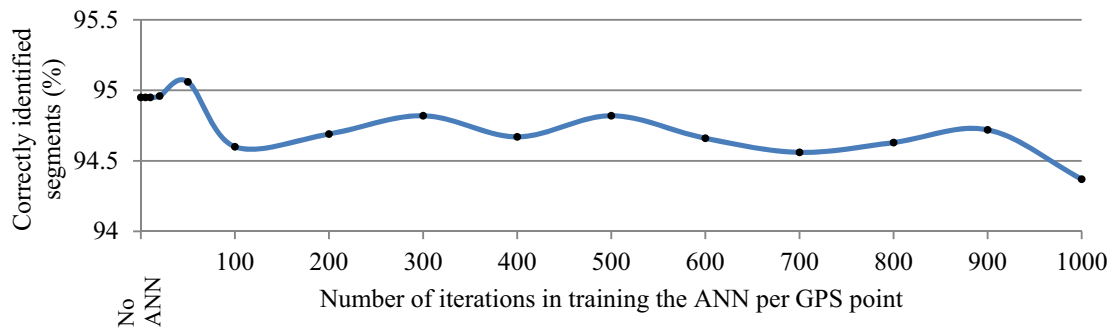
c) Inputs to the ANN: A constant value; Number of hidden units: 20; Training rate: 0.01; Average processing time for one GPS point: 5.07422 millisecond.



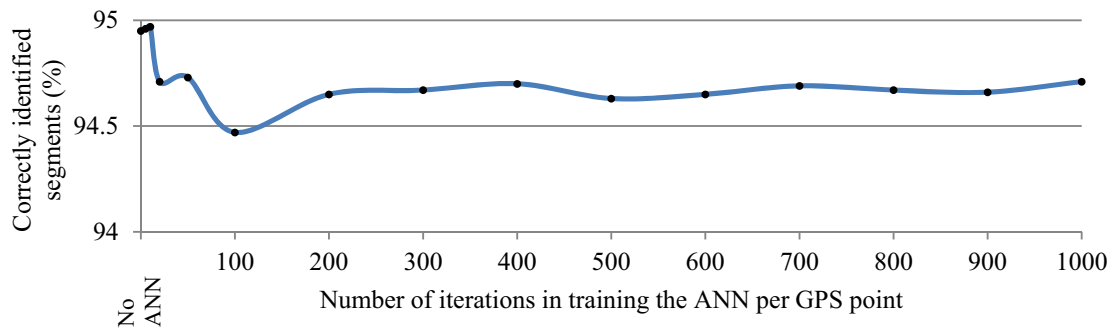
d) Inputs to the ANN: A constant value; Number of hidden units: 20; Training rate: 0.001; Average processing time for one GPS point: 4.99849 millisecond.



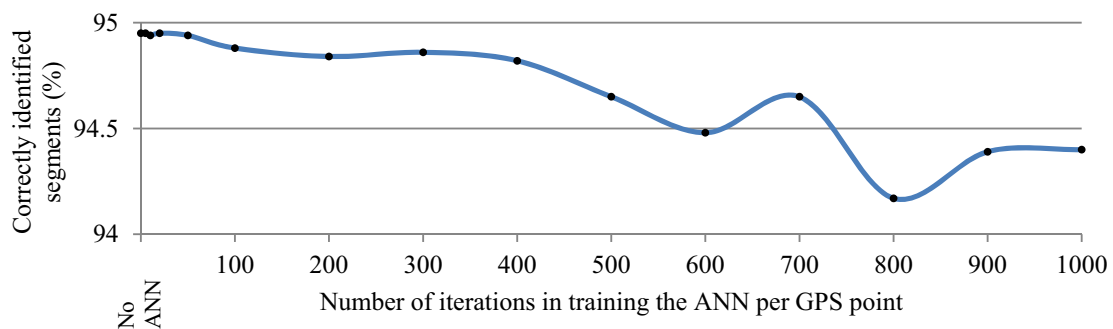
e) Inputs to the ANN: Speed; Number of hidden units: 10; Training rate: 0.01; Average processing time for one GPS point: 4.61981 millisecond.



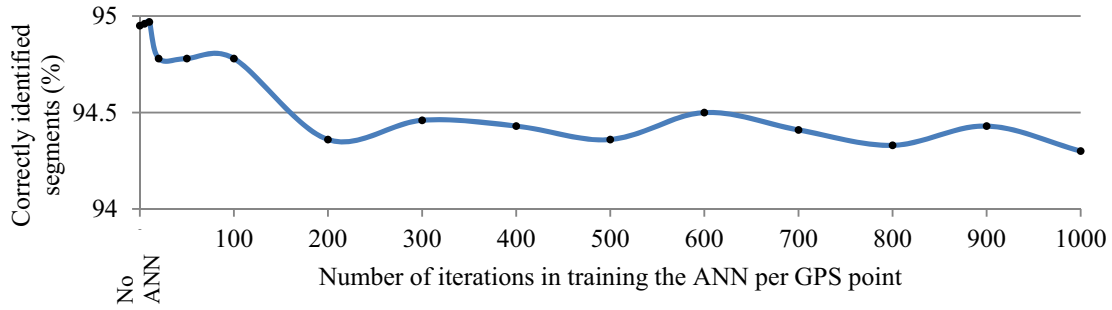
f) Inputs to the ANN: Speed; Number of hidden units: 10; Training rate: 0.001; Average processing time for one GPS point: 4.61981 millisecond.



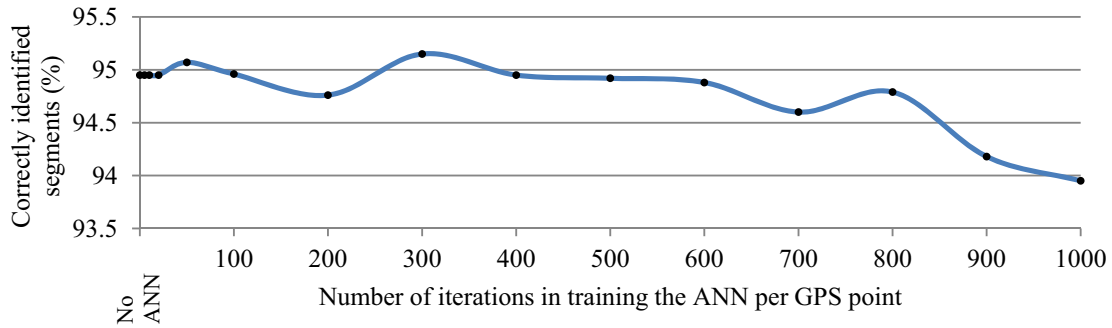
g) Inputs to the ANN: Speed; Number of hidden units: 20; Training rate: 0.01; Average processing time for one GPS point: 5.45289 millisecond.



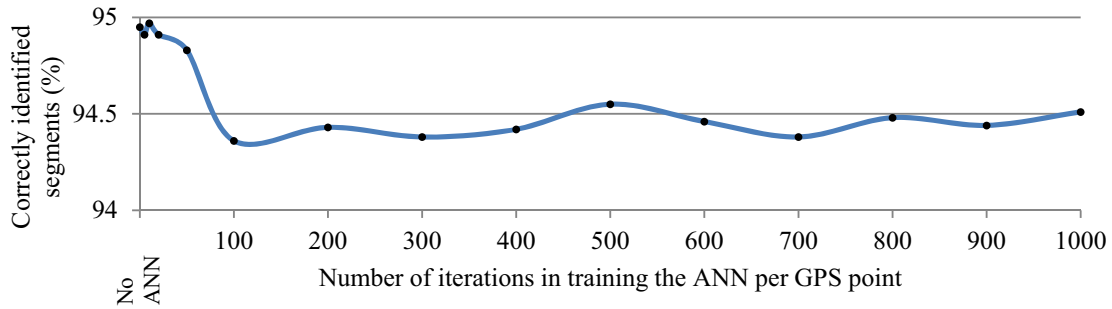
h) Inputs to the ANN: Speed; Number of hidden units: 20; Training rate: 0.001; Average processing time for one GPS point: 5.30142 millisecond.



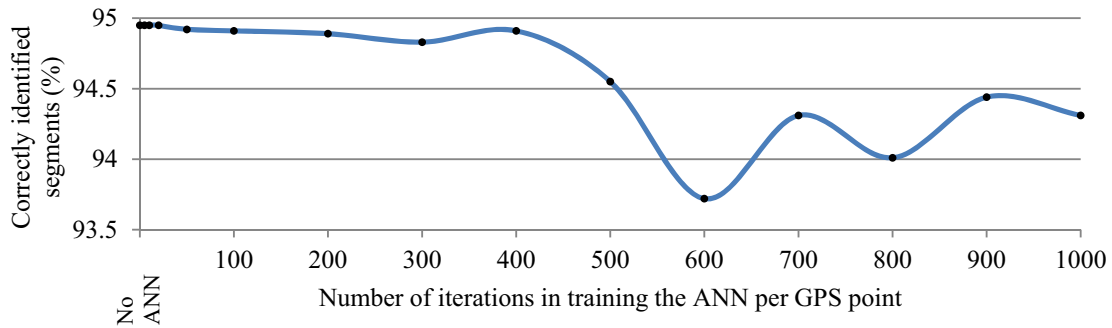
i) Inputs to the ANN: Speed and HDOP; Number of hidden units: 10; Training rate: 0.01; Average processing time for one GPS point: 4.92275 millisecond.



j) Inputs to the ANN: Speed and HDOP; Number of hidden units: 10; Training rate: 0.001; Average processing time for one GPS point: 4.61981 millisecond.



k) Inputs to the ANN: Speed and HDOP; Number of hidden units: 20; Training rate: 0.01; Average processing time for one GPS point: 5.37716 millisecond.



l) Inputs to the ANN: Speed and HDOP; Number of hidden units: 20; Training rate: 0.001; Average processing time for one GPS point: 5.22569 millisecond.

Figure 4. The percentage of correctly identified segments for different numbers of iterations in training the ANN per GPS point.

The worst results are observed when speed and HDOP are considered as inputs to the ANN (see Figures 4 i, j, k, and l). The speed and HDOP were normalized between 0 and 1 before they are inputted to the ANN and speed was negated because its effect on the output is reverse; the faster the vehicle's speed, the less the GPS horizontal error is [1; 25; 14] (the results were worse when the speed was not negated). The percentage of correctly identified segments was improved by 0.07% on average when speed was considered as the only input comparing to when both speed and HDOP were considered as inputs, though it was not better than when no ANN was integrated with the map-matching algorithm (see Figures 4 e, f, g, h). The constant value, as the sole input to the ANN, improved the performance of the map-matching algorithm in identifying the correct segment in all cases comparing to when no ANN was integrated with the map-matching algorithm (see Figures a, b, c, and d). The percentage of correctly identified segments and processing time were increased by 0.65% and 15%, respectively, for cases with a constant value as the input to the ANN on average comparing to the original map-matching algorithm. The obtained better accuracy when only a constant value is used as input to the ANN implies that the ANN's output, i.e., GPS error, is more strongly dependent on location and time rather than speed or HDOP and that the ANN is a spatio-temporally dynamic function.

The best results were obtained when the input to the ANN was only a constant value, the number of hidden units was 20, the training rate was 0.01, and the number of iterations in training the ANN per GPS point was 50. In this case, 96.22% of identified segments were correct which is 1.27% greater than the results of the original map-matching algorithm and the total processing time for 13204 GPS points was increased only less than a second (less than a millisecond for each GPS point). The improvement in correct segment identification, although small, is crucial for map-matching algorithms. The original map-matching algorithm identified a wrong segment for 667 GPS points in situations where it was misled by the GPS positional error in the dense road network. Integrating the ANN enabled the map-matching algorithm to solve 168 of those cases and identify the correct segment for them. Therefore, the small improvement (1.27%) happens on GPS points with relatively large errors, located around intersections or between close parallel roads.

Fig. 5 shows examples of how the ANN improves the coordinates of raw GPS points. The crosses in this figure represent the modified GPS points which reduce the difficulty of finding the correct segment and ease the work of the map-matching algorithm. The examples in this figure show how integrating the ANN with the map-matching algorithm changes the identified segment for a GPS point from wrong to right.

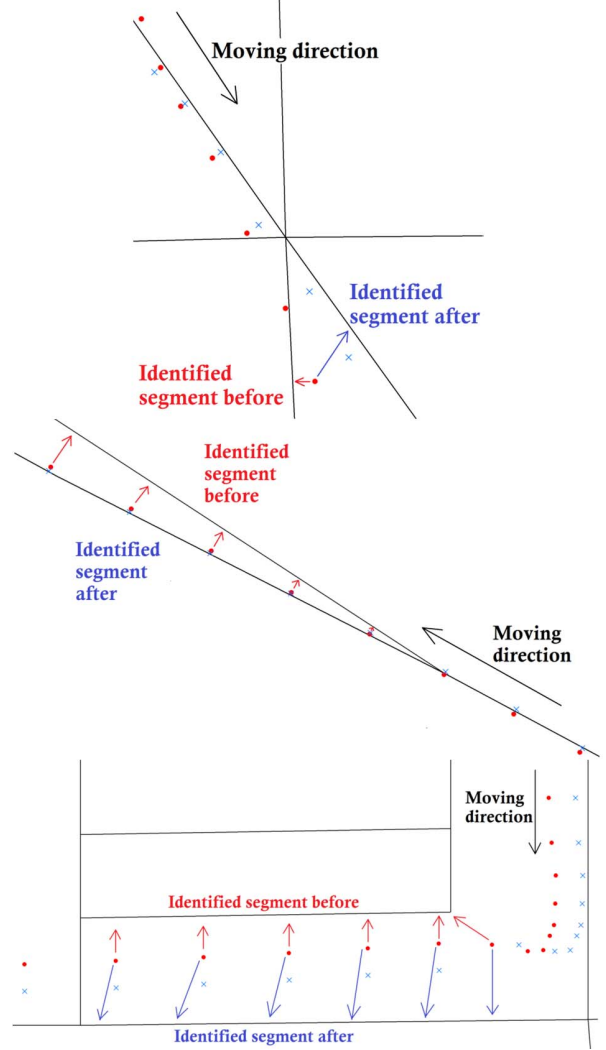


Figure 5. Examples showing improved performance of the map-matching algorithm after integrating it with the ANN: circles represent raw GPS points and crosses represent the modified location of GPS points using the ANN.

The along-track accuracy of the projected GPS points plays an important role in intelligent navigation systems. Since the actual coordinates of the raw GPS points are not known, we cannot measure the actual improvement in the along-track accuracy of the projected GPS points. However, we can measure the difference in the location of the projected GPS points before and after integrating the ANN with the map-matching algorithm. This difference is always along the track. The projected location of each GPS point after integrating the ANN with the map-matching algorithm was 11 cm different than its projected location before the integration on average. Although, 11 cm improvement in the along-track accuracy of the projected GPS points is not significant for navigation systems, it highlights a potential in the proposed approach, requiring further research, which is absent in the original map-matching algorithm.

V. CONCLUSIONS AND FUTURE RESEARCH

Considering that GPS errors are spatio-temporally autocorrelated, detected errors in GPS points can be used to improve the accuracy of proceeding GPS points which in turn improves the chance of correct segment identification in map-matching algorithms. This work validated the possibility of software-based approaches to improve the correct segment identification percentage of map-matching algorithms by improving the horizontal accuracy of GPS points before they are inputted into the algorithms. Although the proposed approach may not be as effective as DGPS or similar approaches requiring additional hardware or outsourced data, it is advantageous in that it does not need any additional hardware or external data and yet has the potential to improve the accuracy of GPS points and the map-matching algorithm. This was shown in this paper by integrating an ANN which is continuously trained by the output of the map-matching algorithm and used to predict the horizontal error for the proceeding GPS points. The percentage of correctly identified segments by the map-matching algorithm was increased from 94.95% to 96.22% and the overall computation time was increased by 15%. This is noteworthy because this improvement happens for GPS points which are located in challenging areas, e.g., around intersections and between close parallel roads, where position, heading, and direction of GPS points lead the original map-matching algorithm to the wrong segment.

It was expected that by considering speed and HDOP as inputs to the ANN, the performance of the map-matching algorithm would improve, but this did not happen, at least based on the experimental results in this work. The reason is that it was observed that the pattern of GPS errors is not as dependent on speed and HDOP as it is on location and time. In other words, this pattern changes gradually as the vehicle moves through streets but it does not change considerably with speed and HDOP. Thus, there is a spatio-temporal autocorrelation among GPS errors which was captured best by the ANN with a constant value as the sole input. Future research will be concentrated on improving the performance of the ANN by investigating spatio-temporal factors affecting GPS errors and considering them as inputs to the ANN.

REFERENCES

- [1] Ochieng, W. Y., Quddus, M., & Noland, R. B. (2004). Map-matching in complex urban road networks. *Brazilian Journal of Cartography (Revista Brasileira de cartografia)*, 55 (2), 1-18.
- [2] InsideGNSS. (2008). New GPS standard positioning system (SPS) performance standard. US Department of Defense.
- [3] Taylor, G., Blewitt, G., Steup, D., Corbett, S., & Car, A. (2001). Road reduction filtering for GPS-GIS Navigation. *Transactions in GIS*, 5 (3), 193-207.
- [4] Karimi, H. A., Liu, X., Liu, S., & Hammad, A. (2004). GPSLoc: Framework for predicting Global Positioning System quality of service. *Journal of computing in civil engineering*, 18 (3), 196-206.
- [5] Ren, M., & Karimi, H. A. (2012a). A fuzzy logic map matching for wheelchair navigation. *GPS solutions*, 16 (3), 273-282.
- [6] Ren, M., & Karimi, H. A. (2012b). Movement pattern recognition assisted map matching for pedestrian/wheelchair navigation. *Journal of Navigation*, 65 (4), 617-633.
- [7] 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*, 10 (3), 103-115.
- [8] Quddus, M. A., Noland, R. B., & Ochieng, W. Y. (2005). Validation of map matching algorithms using high precision positioning with GPS. *Journal of Navigation*, 58 (2), 257-271.
- [9] Srinivasan, D., Cheu, R. L., & Tan, C. W. (2003). Development of an improved ERP system using GPS and AI techniques. In *Proceedings of Intelligent Transportation Systems Conference*. 1, pp. 554-559. IEEE.
- [10] Jagadeesh, G. R., Srikanthan, T., & Zhang, X. D. (2004). A map matching method for GPS based real-time vehicle location. *Journal of Navigation*, 57 (3), 429-440.
- [11] Yang, J.-s., Kang, S. P., & Chon, K.-s. (2005). The map matching algorithm of GPS data with relatively long polling time intervals. *Journal of the Eastern Asia Society for Transportation Studies*, 6, 2561-2573.
- [12] Zhao, L., Ochieng, W. Y., Quddus, M. A., & Noland, R. B. (2003). An extended Kalman filter algorithm for integrating GPS and low-cost dead reckoning system data for vehicle performance and emissions monitoring. *Journal of Navigation*, 56 (2), 257-275.
- [13] Chen, W., Yu, M., Li, Z. L., & Chen, Y. Q. (2003). Integrated vehicle navigation system for urban applications. In *Proceedings of the International Conference on Global Navigation Satellite Systems (GNSS)*, (pp. 15-22). European Space Agency, Graz, Austria.
- [14] Hashemi, M., & Karimi, H. A. (2016). A weight-based map-matching algorithm for vehicle navigation in complex urban networks. *Journal of Intelligent Transportation Systems*, DOI: 10.1080/15472450.2016.1166058.
- [15] Ren, M., & Karimi, H. A. (2009). A chain-code-based map matching algorithm for wheelchair navigation. *Transactions in GIS*, 13 (2), 197-214.
- [16] Hashemi, M., & Karimi, H. A. (2014). A critical review of real-time map-matching algorithms: Current issues and future directions. *Computers, Environment and Urban Systems*, 48, 153-165.
- [17] Greenfeld, J. S. (2002). Matching GPS observations to locations on a digital map. In *Proceedings of the 81th Annual Meeting of the Transportation Research Board*. Washington D.C.
- [18] White, C. E., Bernstein, D., & Kornhauser, A. L. (2000). Some map matching algorithms for personal navigation assistants. *Transportation Research Part C: Emerging Technologies*, 8 (1), 91-108.
- [19] Quddus, M. A., Ochieng, W. Y., Zhao, L., & Noland, R. B. (2003). A general map matching algorithm for transport telematics applications. *GPS Solutions*, 7 (3), 157-167.
- [20] Hu, G. R., Khoo, V. H., Goh, P. C., & Law, C. L. (2002). Internet-based GPS VRS RTK positioning with a multiple reference station network. *Journal of Global Positioning Systems*, 1 (2), 113-120.
- [21] Retscher, G. (2002). Accuracy performance of virtual reference station (VRS) networks. *Journal of Global Positioning Systems*, 1 (1), 40-47.
- [22] Witte, T. H., & Wilson, A. M. (2005). Accuracy of WAAS-enabled GPS for the determination of position and speed over ground. *Journal of Biomechanics*, 38 (8), 1717-1722.
- [23] Socharoentum, M., & Karimi, H. A. (2011). Collaborative map matching in Nav2Nav. In *Proceedings of 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)* (pp. 251-257). Orlando, Florida, USA: IEEE.
- [24] Hasan, A. M., Samsudin, K., Ramli, A. R., Azmir, R. S., & Ismaeel, S. A. (2009). A review of navigation systems (integration and algorithms). *Australian Journal of Basic and Applied Sciences*, 3 (2), 943-959.

- [25] Taylor, G., Brunsdon, C., Li, J., Olden, A., Steup, D., & Winter, M. (2006). GPS accuracy estimation using map-matching techniques: Applied to vehicle positioning and odometer calibration. *Computers, Environment and Urban Systems* , 30 (6), 757-772.
- [26] Ren, M. (2012). Advanced map matching technologies and techniques for pedestrian/wheelchair navigation, PhD Dissertation. Pittsburgh, Pennsylvania, USA: School of Information Sciences, University of Pittsburgh.
- [27] Kubrak, D., Macabiau, C., Monnerat, M., & Bouchert, M. L. (2006). Vehicular navigation using a tight integration of aided-GPS and low-cost MEMS sensors. *Proceedings of the Institute of Navigation Technical Meeting*, (pp. 149-160).
- [28] Velaga, N. R., Quddus, M. A., & Bristow, A. L. (2009). Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. *Transportation Research Part C: Emerging Technologies* , 17 (6), 672-683.
- [29] Velaga, N. R., Quddus, M. A., & Bristow, A. L. (2012). Improving the performance of a topological map-matching algorithm through error detection and correction. *Journal of Intelligent Transportation Systems* , 16 (3), 147-158.
- [30] Quddus, M. A., Ochieng, W. Y., & Noland, R. B. (2006a). Integrity of map-matching algorithms. *Transportation Research Part C: Emerging Technologies* , 14 (4), 283-302.
- [31] Syed, S., & Cannon, M. E. (2004). Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons. In *proceedings of the Institute of Navigation (ION) national technical meeting*, (pp. 26-28). San Diego, CA.
- [32] Mosavi, M. R. (2006). Comparing DGPS corrections prediction using neural network, fuzzy neural network, and Kalman filter. *GPS Solutions* , 10 (2), 97-107.
- [33] Kröse, B., Krose, B., Smagt, P. v., & Smagt, P. (1996). *An introduction to neural networks* (8th Edition ed.). Amsterdam: University of Amsterdam.
- [34] Fausett, L. (1994). *Fundamentals of neural networks: architectures, algorithms and applications*. Englewood Cliffs, New Jersey: Prentice Hall.
- [35] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature* , 323 (9), 533-536.