A Map Matching Method for GPS Based Real-Time Vehicle Location

G. R. Jagadeesh, T. Srikanthan and X. D. Zhang

(*Nanyang Technological University*, *Singapore*)
(Email: jagadeesh@pmail.ntu.edu.sg)

Accurate vehicle location is essential for various applications in the field of intelligent transportation systems (ITS). Existing vehicle location systems rely on multiple positioning sensors and powerful computing devices to execute complex map matching algorithms. There exists a strong need for exploring a solution for vehicle location that relies on a GPS receiver as the sole means of positioning and does not require complex computations. Towards this end, the error characteristics of the GPS signal were studied through the analysis of GPS data collected during test drives. Based on the inferences drawn and a simple fuzzy rule set, a novel yet simple map matching algorithm was developed. Due to the difficulties in testing the algorithm through on-road trials, a simulation environment that is capable of reproducing the field conditions in the laboratory was developed. Simulation results confirm that the proposed algorithm overcomes many of the inadequacies of the existing methods and is capable of achieving high accuracy with minimal computational requirements.

KEY WORDS

1. Road Network.   2. Vehicle Location.   3. GPS.   4. Map Matching.

1. INTRODUCTION.   Accurately locating a vehicle on a map is of crucial importance to various applications in the field of intelligent transportation systems (ITS). Automatic vehicle location (AVL) systems enable transit fleet operators to save millions of dollars through reduction of fuel wastage and better utilization of resources. Dynamic route guidance [1] is another key component of ITS that helps to minimize traffic congestion and improve the overall utilization of the road network by assisting drivers to avoid congested routes and guiding them in an unfamiliar locality. Applications such as the ones mentioned above are predicated on the ability to locate the mobile user on a road network map.

A positioning system is indispensable for vehicle location. One of the oldest positioning techniques is dead reckoning, which is based on the idea that it is possible to calculate the vehicle's position at any instance provided the starting location and all previous displacements are known. Dead reckoning incrementally integrates the distance travelled and the direction of travel relative to a known location. Information regarding the displacement and the direction are typically obtained from an onboard odometer sensor and gyroscope respectively. A major drawback of the dead reckoning method is that if the errors associated with the previous measurements are not properly compensated, then these errors will accumulate as the vehicle continues

to travel. Also, it may not be convenient to use dead reckoning along with non-vehicle-based handheld personal navigation assistants (PNA) [2] that are becoming increasingly popular.

The Global Positioning System (GPS) [3] is a satellite-based radio navigation system designed and funded by the US Department of Defense (DoD). It provides people with answers to the age-old question 'Where on earth am I?' In the past, the GPS signal was artificially degraded by the DoD through a process known as selective availability in order to limit the positional accuracy available to civilian users. With selective availability operational, the position estimated by the GPS (in terms of latitude and longitude) is within 100 m of the 'true position' 95% of the time [4]. This level of accuracy is insufficient for correctly locating a vehicle on a dense road network. For this reason, GPS was seldom used as the sole means of positioning in vehicle location systems in the past. Instead it used to be supplemented by another positioning method such as dead reckoning.

The US Government removed selective availability on 1 May 2000 resulting in a substantial increase in the accuracy of the GPS. Positional accuracy of the order of 10–20 m is now possible subject to the availability of signals from at least 4 satellites [4]. However, in dense urban environments, high-rise buildings and other tall structures often block the satellites from the direct 'line of sight' of the GPS receiver resulting in further degradation of accuracy and in some cases, temporary outage of the GPS signal. Nevertheless, the relative improvement in the accuracy of the GPS caused by the removal of selective availability and the demand for stand-alone and portable PNAs serve as compelling reasons for attempting to realize a vehicle location method that uses GPS as the sole means of positioning.

The positional output of the GPS in terms of latitude and longitude can be converted into X and Y coordinates with respect to a two-dimensional surface. However, for most applications that require vehicle location, such a positional output is meaningless unless it can be correlated to a road. Also, the errors inherent in the output of the positioning system can be corrected by considering the fact that the vehicle is mostly constrained to a finite network of roads. For instance, an inaccurate GPS output that implies that the vehicle is in the middle of a building or a lake can immediately be ruled as incorrect. Map matching algorithms match the inaccurate raw position provided by the positioning system to a position on the road network by comparing the trajectory of the vehicle with the shapes of the roads in the network.

In the following sections, we examine the existing map matching methods and identify the drawbacks associated with them. Subsequently, we present an efficient solution for map matching that is based on an empirical study and analysis of GPS signal behavior observed during on-road tests. Finally, the performance of the algorithm is evaluated using a novel simulation environment and experimental results are presented.

2. EXISTING APPROACHES TO MAP MATCHING.

Map matching is essentially a problem of comparing path geometries. In a typical digital road map, the roads in the network are represented as piecewise linear curves. A simple and intuitive way of performing map matching is to match the raw position estimated by the positioning system to the nearest road segment [2]. While this method, known as point-to-arc mapping, is easy to implement it has many disadvantages.
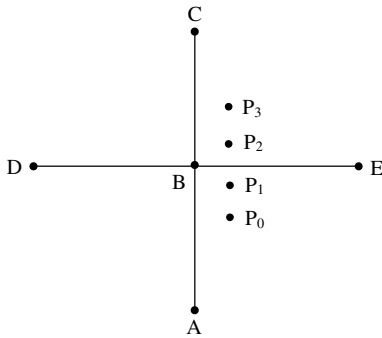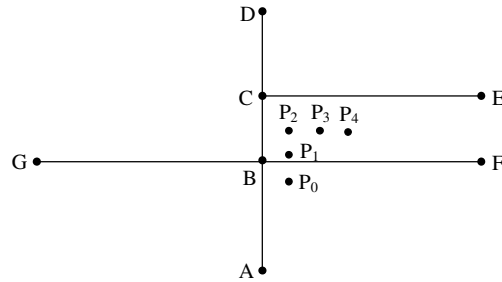
Figure 1. Point-to-arc map matching.

Figure 2. Arc-to-arc map matching.

Firstly, it does not make use of historical information about the vehicle's motion and this can result in mismatches similar to the case illustrated in Figure 1. The sequence of estimated positions P_0, P_1, P_2 and P_3 indicate that the vehicle is travelling along the path comprising of road segments AB and BC. However, since the previously estimated positions are not taken into consideration, the point-to-arc method, the points P_1 and P_2 would be wrongly matched to the closest road segment BE. In route guidance systems, such mismatches are likely to result in wrong guidance instructions being issued to the driver. Another shortcoming of the point-to-arc method is that the matched position can oscillate between two closely situated parallel roads especially in dense urban street networks.

Some of the drawbacks of the point-to-arc method can be overcome by another geometric method known as arc-to-arc mapping. The arc-to-arc method matches the piecewise linear curve formed by the sequence of estimated positions to a piecewise linear curve corresponding to a path in the network based on their closeness [2] or similarity [5]. Results of field trials provided in [2] indicate that while the arc-to-arc method consistently outperforms a simple point-to-arc method, it fails to outperform a point-to-arc method that makes use of information about the vehicle heading. This is possibly because arc-to-arc mapping can perform poorly in situations like the one illustrated in Figure 2, where the piecewise linear curve formed by points P_0, P_1, P_2, P_3 and P_4 is equally close and similar to two candidate paths namely ABF and BCE.

Algorithms that make use of the geometric approach can be improved by incorporating topological information such that only those road segments that are directly connected to the current road of travel are considered. However, this can result in one wrong match leading to a sequence of wrong matches. Another class of algorithms, known as probabilistic algorithms, makes use of statistical error models of the positioning sensor to define a confidence region within which the true vehicle position may lie. Only those roads that lie within this region are considered for map matching. Although these algorithms can recover from wrong matches quickly, they require more computation time. In dense urban road networks, it is often difficult to precisely identify the road on which the vehicle is travelling. Rather, the map-matching algorithm may only be able to determine that the vehicle is more likely to be on some roads and less likely to be on certain other roads. Fuzzy logic has been proven to be an effective way to deal with such ambiguous situations. Because map matching is a decision making process involving a degree of ambiguity, fuzzy logic based algorithms are often employed to perform map matching [1].

In this work, we propose an efficient map-matching algorithm that can be implemented on systems with low computational resources such as a handheld device. We conceptually split the map matching problem into two tasks namely, correctly identifying the road of travel and, determining the exact location of the vehicle on the identified road. We make use of a simple fuzzy rule based inference system for identifying the road of travel. For matching the GPS-estimated position of the vehicle onto the identified road, we employ a novel method that was devised on the basis of an empirical study of the GPS error characteristics presented in the following section.

3. GPS ERROR CHARACTERISTICS AND THEIR IMPLICATIONS FOR MAP MATCHING. We envisage that the accuracy of a map-matching algorithm that relies solely on GPS can be improved by making use of knowledge about GPS signal behaviour under actual field conditions. Towards this end, a vehicle equipped with a SiRF GPS receiver was driven in and around the Nanyang Technological University campus in Singapore. The positional data in terms of latitude and longitude were converted to coordinates in the two-dimensional coordinate system and logged using a portable computer at the rate of 1 reading per second. For the purpose of the following discussion, we use the term GPS error to denote the horizontal distance between the position estimated by the GPS and the true position of the vehicle. Since it is difficult to measure the true position of a moving vehicle at any given instant, we had to settle for an approximation. For each position estimated by the GPS, the closest point on the road centerlines that form part of the test route was assumed to be the true position of the vehicle.

The results of the data analysis are as follows. The GPS error was found to be in the range 0–15 m. However, it is not advisable to draw any conclusions from the above observation due to two reasons. Firstly, the error represents the distance between the GPS-estimated position and the road centerline and not the true position of the vehicle, which is difficult to determine. Secondly, the test was conducted in a sparse road network in the outskirts of the city where high-rise buildings and other factors that can cause the GPS performance to deteriorate are absent. The GPS error is likely to be higher in dense urban areas. However, there are a couple of other observations that are useful for map matching.

Observation 1. Although the GPS error ranged from 0 to 15 m, the error difference between two consecutive readings was found to be much smaller. Other researchers who have performed similar trials with a different GPS receiver have also confirmed that the error difference between consecutive measurements is much smaller than the error itself [6]. In our experiments, the error difference remained less than 2 m for over 95% of the cases. A steep change in error between consecutive readings occurs very rarely and is usually when the GPS receiver suddenly fails to receive signals from the required minimum number of satellites or when it regains them.

Observation 2. The data analysis reveals that, in most cases, the distance between the current and previous GPS-estimated positions is almost equal to the distance between the corresponding true positions of the vehicle. In over 95% of the cases, the difference between the two distances was less than 1 m. Here again, the exceptions are caused by the sudden loss or gain of the required number of satellites.
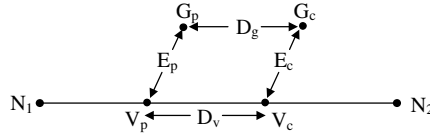
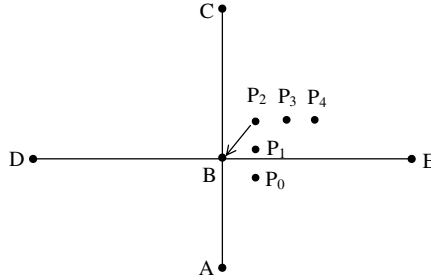Figure 3. GPS error characteristics for a moving vehicle.

Figure 4. Error correction based on GPS heading information.

Let us examine the relevance of the above two observations to the map matching problem with the help of Figure 3. Let N_1N_2 be the road segment on which the vehicle is identified to be traveling. Let V_c and V_p be the current and previous true positions of the vehicle respectively. Let G_c and G_p be the current and previous GPS-estimated positions. E_c and E_p represent the current and previous GPS errors. D_g denotes the distance between the current and previous GPS-estimated positions and D_v represents the distance between the current and previous true positions of the vehicle. Observations 1 and 2 suggest that under normal GPS operating conditions, $E_c \approx E_p$ and $D_g \approx D_v$. This implies that the points V_p, G_p, G_c and V_c form a rough parallelogram.

The information above makes it possible to relatively accurately determine the true position of the moving vehicle corresponding to a position determined by the GPS if the previous GPS-estimated position and the previous true vehicle position are known. This is done simply by finding the fourth vertex of the parallelogram, without the need for any complex computations. The true vehicle position thus determined may not fall exactly on the road because of the 'roughness' of the parallelogram and the inaccuracies of the digital road network. Therefore, the point on the road network that is closest to the fourth vertex of the parallelogram can be identified as the true position of the vehicle. We shall refer to this method of correlating the GPS-estimated position to a position on the road of travel as the parallelogram method.

Another inference from the analysis above is that under normal GPS operating conditions, the direction of motion or heading indicated by two consecutive GPS-estimated positions is almost similar to the actual heading of the vehicle, which is the same as the heading of the road segment in which the vehicle travels. Thus, the heading information obtained from the GPS readings can be used to determine the likelihood of a given road being the actual road of travel. It can also help to correct any accumulated map matching errors and reconcile the true position of the vehicle from time to time. For instance, in Figure 4 the heading information obtained from
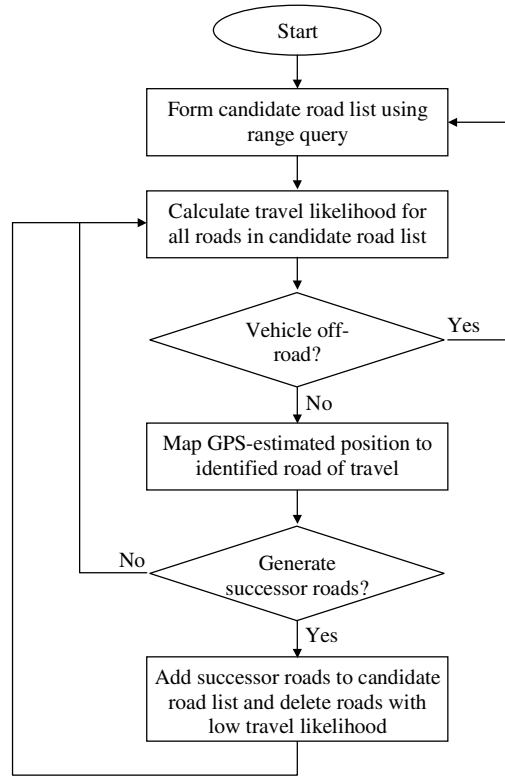
```
        ┌─────────────┐
        │    Start    │
        └──────┬──────┘
               │
               ▼
  ┌────────────────────────────┐
  │ Form candidate road list   │◄──────────┐
  │ using range query          │           │
  └──────────────┬─────────────┘           │
                 │                          │
                 ▼                          │
  ┌────────────────────────────┐           │
  │ Calculate travel likelihood │◄───┐     │
  │ for all roads in candidate  │    │     │
  │ road list                   │    │     │
  └──────────────┬─────────────┘    │     │
                 │                   │     │
                 ▼                   │     │
          ╱─────────────╲     Yes    │     │
         ╱ Vehicle off-  ╲───────────┼─────┘
         ╲ road?         ╱           │
          ╲─────────────╱            │
                 │ No                │
                 ▼                   │
  ┌────────────────────────────┐    │
  │ Map GPS-estimated position │    │
  │ to identified road of      │    │
  │ travel                     │    │
  └──────────────┬─────────────┘    │
                 │                   │
                 ▼                   │
    No    ╱─────────────╲            │
  ┌───────╲ Generate     ╱           │
  │        ╲ successor   ╱           │
  │         ╲ roads?    ╱            │
  │          ╲────────╱              │
  │              │ Yes               │
  │              ▼                   │
  │  ┌────────────────────────┐      │
  │  │ Add successor roads to │      │
  │  │ candidate road list    │      │
  │  │ and delete roads with  │      │
  │  │ low travel likelihood  │      │
  │  └───────────┬────────────┘      │
  │              │                   │
  └──────────────┴───────────────────┘
```

Figure 5. The map-matching algorithm.

the GPS-estimated positions P_1, P_2 and P_3 indicates that the vehicle has just made a 90-degree turn onto the road segment BE. Hence it can be assumed with reasonable accuracy that the true position of the vehicle corresponding to P_2 is the point B and the GPS-estimated position P_3 can be matched to the road segment BE using the parallelogram method.

4. THE MAP MATCHING ALGORITHM.   The proposed map-matching algorithm is presented as a flow chart in Figure 5. In the figure and the discussion that follows, the term road denotes a road segment that lies between two intersections or an intersection and a dead end. In order to minimize the amount of computation required, we do not attempt to compare the trajectory of GPS-estimated positions to each road in the network. Instead, we limit the computations to a small number of roads that are maintained in a candidate road list. As the vehicle moves along the road network, the contents of the candidate road list are updated if one of the following two situations occurs.

    The first situation is when the vehicle is off-road. A vehicle may occasionally leave the road network to enter a car park or fuel station. An off-road condition can also occur when the vehicle enters a road that is not represented in the digital map. In such cases, the candidate road list is continuously updated by assigning to it all the roads that lie within a certain distance of the current GPS-estimated position of the vehicle.

(a) Truth value of 'heading
difference is small'

(b) Truth value of 'distance
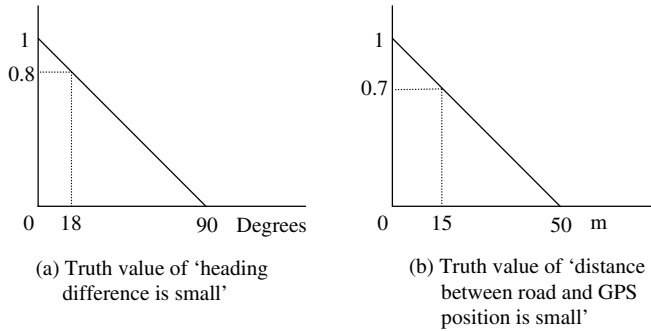between road and GPS
position is small'

Figure 6. Membership functions for Rule 1.

We refer to this process as range query. In the proposed algorithm, the range was chosen to be twice that of the error range of the GPS receiver. In addition to the normal off-road conditions, the vehicle is assumed to be off-road at system startup and a new candidate road list is created by performing a range query. The second situation in which the candidate road list is updated is when the vehicle has travelled a significant portion of the road that is identified as the current road of travel. In that case, the successor roads that are topologically connected to the upcoming intersection are added to the candidate road list. The older roads in the candidate road list, which are less likely to be the road of travel, are removed from the list.

The algorithm attempts to evaluate the likelihood of each road in the candidate road list being the actual road of travel. For each road, a factor known as resemblance is calculated based on two factors namely the similarity of the road heading to the heading obtained from the previous and current GPS-estimated positions and the closeness of the road to the current GPS-estimated position. A simple fuzzy rule (Rule 1) is used for this purpose. In Rule 1, the phrase 'heading difference' denotes the difference between the heading of the road and the heading obtained from two consecutive GPS-estimated positions.

Rule 1.   IF the heading difference is small AND the distance between the road and the current GPS-estimated position is small THEN resemblance of the road is high.

The linguistically vague expressions in the above rule such as 'heading difference is small' and 'distance between the road and the current GPS position is small' are represented using fuzzy sets. Each such expression has a degree of truth associated with it. This degree of truth or truth value ranges from 0 to 1 and is characterized by membership functions similar to the ones illustrated in Figure 6. Referring to Rule 1, the AND (intersection) operation is implemented using the function $min(x, y)$, which returns the smallest of the two operands. As an example, let us assume that a road is 15 m away from the current GPS estimated position and the heading difference is $18°$. Membership functions shown in Figure 6(a) and 6(b) will yield truth values of 0·8 and 0·7 respectively. Applying the AND operator, the truth value of the resemblance being high for that road is 0·7. More information on fuzzy logic concepts and their application to similar practical situations can be found in [7].

While it is intuitive to identify the road with the highest resemblance value as the current road of travel, doing so can occasionally lead to spurious identifications. For instance, a steep change in the GPS error due to a sudden loss or gain of satellites may

result in a wrong road being momentarily assigned a high resemblance value. In order to avoid such momentary wrong identifications, the algorithm considers the last two resemblance values of a given road to determine the likelihood of it being the current road of travel. A factor known as travel likelihood is calculated for each road based on Rule 2 given below. The road with the highest travel likelihood is identified as the road of travel.

Rule 2.   IF the previous resemblance of the road is high AND the current resemblance is high THEN the travel likelihood of the road is high.

The algorithm continuously checks if the vehicle is off the road network. An off-road condition is detected when all the roads in the candidate list have very low resemblance values. A suitable threshold for resemblance can be fixed based on the error range of the GPS receiver and other relevant factors. Here again, two consecutive measurements are taken into account in order to prevent spurious detection of off-road conditions. Rule 3 is used for this purpose. Successor roads are added to the candidate road list if the vehicle has covered at least 25% of the road that is identified as the current road of travel.

Rule 3.   IF the previous maximum resemblance of all the roads in the candidate road list is low AND the current maximum resemblance of all the roads in the list is low THEN the vehicle is off-road.

The heading information obtained from the GPS-estimated positions is meaningful only if the vehicle is in motion. However, during the course of the journey, the vehicle is likely to stop many times at traffic signals or move at a very slow pace due to traffic congestion. In such situations, the GPS heading information may be inaccurate and it should not be considered for calculating the resemblance. Thus when the vehicle is stationary or almost stationary, the resemblance of the road is computed solely based on its closeness to the GPS-estimated position. This can result in a wrong road being identified as the road of travel. To prevent this the algorithm prohibits a new road being assigned as the current road of travel when the vehicle is not in motion.

Once the current road of travel is determined by the algorithm in the manner described above, the current GPS-estimated position is matched to the identified road using the parallelogram method. The parallelogram method requires the previous GPS-estimated position and the previous map-matched position, which should ideally be the same as the previous true position of the vehicle. However, when the vehicle enters a road for the first time after an off-road condition, the previous map-matched position is not known. In such situations, the parallelogram method is not used and the GPS-estimated position is matched to closest point on the road. Any error that might result from this approximation can be corrected when the vehicle makes a turn at an intersection as illustrated in Figure 4.

5. PERFORMANCE OF THE ALGORITHM. In order to evaluate the performance of the algorithm, we make use of a metric known as the correct road-matching ratio, which is the percentage of correct matches produced by the
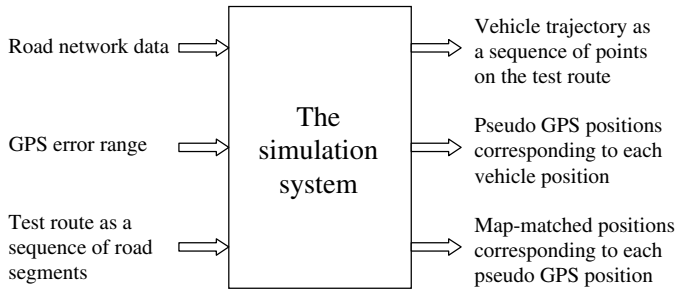
Figure 7. The simulation environment.

algorithm. A match is deemed to be correct if the road to which the GPS-estimated position is matched is indeed the actual road being traversed by the vehicle.

It is worth recalling that the map-matching algorithm waits for two measurements before confirming any road as the current road of travel. Also, to prevent spurious road identifications, the algorithm does not identify a new road of travel when the vehicle is either stationary or moving at a very slow speed, which is often the case at intersections. This will result in a situation wherein the vehicle has just entered a new road at an intersection but the previously traversed road is identified by the algorithm as the current road of travel. Such cases would be considered as a wrong matching for the purpose of calculating the correct road-matching ratio. However, this is a harsh decision since the map-matched position still lies along the actual trajectory of the vehicle. Also, a delay of one or two seconds in confirming the new road of travel would not interfere with the operation of services that require vehicle location such as route guidance. Therefore, we also calculate another evaluation metric known as the correct route-matching ratio, in which a match is deemed to be correct as long as the map-matched position lies along the actual route of the vehicle.

5.1. *The evaluation set-up.*   The ideal way of evaluating the performance of the map-matching algorithm is to conduct test drives on real road networks. However, one major disadvantage of real-road tests is that it is very difficult to determine the true position of a moving vehicle continuously. Also, to identify possible flaws in the algorithm and to test the effectiveness of the improvements made, repeated trials are necessary. Conducting a large number of on-road trials is costly in terms of time, effort and money. Therefore, we decided to develop a software-based simulation environment that is capable of reproducing on-road conditions in the laboratory itself. This involves simulating the motion of the vehicle along a specified route and generating the sequence of pseudo GPS positions corresponding to the vehicle trajectory.

The simulation environment was developed using Borland C++ Builder. For the simulation system to be realistic and accurate, it needs to mimic the real on-road conditions as closely as possible. Towards this end, the acceleration/deceleration pattern of the simulated moving vehicle was based on a variety of factors such as road type, distance to the next intersection, presence of sharp curves etc. The pseudo GPS positions were generated based on the inferences drawn from the analysis of real GPS data collected during on-road test drives as explained in Section 3. The map-matching algorithm was integrated into the simulation system and was tasked to match each pseudo GPS position onto the road network. The inputs and outputs of the simulation system are illustrated in Figure 7.

Table 1. Simulation results for the sparse network with low GPS error.

Route No.	Average number of GPS readings	Correct road matching ratio (%)	Correct route matching ratio (%)
1	6452	98·22	99·35
2	4273	95·88	99·25
3	4726	96·00	99·37
4	3484	96·70	98·71
5	7669	96·77	99·48
Average	**5321**	**96·71**	**99·23**

Table 2. Simulation results for the sparse network with high GPS error.

Route No.	Average number of GPS readings	Correct road matching ratio (%)	Correct route matching ratio (%)
1	6541	96·96	98·41
2	4258	94·15	98·97
3	4673	95·06	98·48
4	3423	94·77	96·41
5	7429	95·84	99·11
Average	**5265**	**95·36**	**98·28**

Table 3. Simulation results for the dense network with low GPS error.

Route No.	Average number of GPS readings	Correct road matching ratio (%)	Correct route matching ratio (%)
1	5286	93·64	99·17
2	5503	92·17	98·73
3	4879	92·03	99·30
4	5253	92·88	99·14
5	5085	94·67	99·31
Average	**5201**	**93·08**	**99·13**

The simulations were performed on two networks that are considerably different from each other. The first road network is a sparse one comprising our university campus and the residential areas close to it. The other one is a very dense road network in the centre of the Singapore city. For each road network, simulations were performed with two different GPS error ranges corresponding to low (0–15 m) and high (0–30 m) GPS error conditions. Five test routes were selected in each road network. Each route was simulated 25 times in order to estimate the average performance of the algorithm.

5.2. *Experimental results.* The results obtained from the four sets (sparse network – low error, sparse network – high error, dense network – low error and dense network – high error) of simulations are presented in Tables 1–4. A screen capture corresponding to the dense network – high error category is shown in Figure 8, where the sequence of dots for the pseudo GPS and map-matched positions and the

Table 4. Simulation results for the dense network with high GPS error.

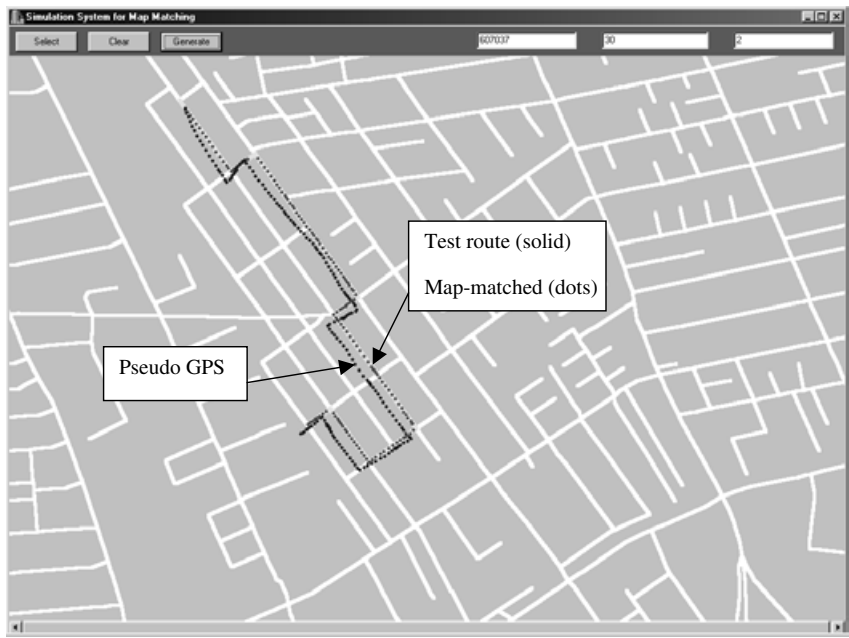| Route No.	Average number of GPS readings	Correct road matching ratio (%)	Correct route matching ratio (%)
1	5216	91·43	95·07
2	5551	87·41	93·30
3	4895	89·05	97·98
4	5200	87·77	97·25
5	5077	92·12	96·73
Average	**5188**	**89·57**	**96·07**

Figure 8. A sample screenshot of the simulation system.

test route are highlighted. (N.B., the map-matched positions coincide with the test route.)

The results establish the predictable fact that for any given road network, the map-matching algorithm consistently performs better under low (0–15 m) GPS error conditions compared to high (0–30 m) GPS error conditions. Also, for a given GPS error condition, the correct road matching ratio and the correct route-matching ratio are consistently better for the sparse network compared to the dense network. For the sparse network, while the average correct road-matching ratio is over 95% for both error conditions, the average correct route-matching ratio is over 98%. The corresponding figures for the dense network were 89% and 96%. The worst case performance (87% correct road matching ratio and 93% correct route matching ratio) was registered by Route 2 in the dense network under high error conditions.

It was not possible for us to benchmark the performance of the algorithm against other methods proposed in the literature due to the scarcity of published data and the absence of uniform testing conditions. However, the results confirm that the algorithm is capable of achieving relatively high accuracy without the need for complex computations. We are confident that the on-field performance of the algorithm would be almost similar to the simulated results. This is because the simulation system was developed based on our study of the real GPS data collected through on-road tests. However, the robustness of the proposed map-matching algorithm in harsh urban conditions, where the GPS performance is likely to deteriorate has to be studied and evaluated in detail.

6. CONCLUSIONS. Applications and services that require accurate vehicle location are likely to become prevalent in the near future. There is a strong need and justification for real-time vehicle location systems that do not rely on multiple positioning sensors and powerful computers. In this work, we have analyzed the GPS signal behavior in order to realize a map-matching algorithm for vehicle location that uses GPS as the sole means of positioning. The proposed algorithm overcomes many of the inaccuracies inherent in existing techniques and is simple enough to be implemented on a handheld computer interfaced to a GPS receiver. The algorithm has been evaluated using a custom-built simulation environment that reproduces real on-road conditions. Experimental results confirm that the algorithm is capable of achieving relatively high accuracy. Since accurate vehicle location is a crucial requirement in services such as route guidance, it would be beneficial to explore the possibility of further improving the performance of the algorithm without the need for any additional resources.

REFERENCES

[1] Y. Zhao, ''Vehicle Location and Navigation Systems'', *Artech House*, 1997.
[2] C. E. White, D. Bernstein and A. L. Kornhauser, ''Some Map Matching Algorithms for Personal Navigation System'', *Transportation Research Part C*, Vol. 8, 2000.
[3] B., Hofmann-Wellenhof, ''Global Positioning System: Theory and Practice'', *Springer-Verlag*, 1997.
[4] W. Y. Ochieng and K. Sauer, ''Urban Road Transport Navigation: Performance of the Global Positioning System after Selective Availability'', *Transportation Research Par C*, Vol. 10, 2002.
[5] R. R. Joshi, ''A New Approach to Map Matching for In-Vehicle Navigation Systems: The Rotational Variation Metric'', *IEEE Intelligent Transportation Systems Conference Proceedings*, 2001.
[6] S. H. Pen and C. S. Tsui, ''Map Matching Algorithm of GPS Vehicle Navigation System'', *Asian Conference on Remote Sensing*, 1998.
[7] K. Tanaka, ''An Introduction to Fuzzy Logic for Practical Applications'', *New York: Springer*, 1997.