Paper number EU-TP0615

Advanced topological map matching: a step towards full integrity and reliability in road transport environments

Lauri Koponen[1*], Rui Sun[2]
1. TomTom Navigation, the Netherlands
2. TomTom Navigation, the Netherlands
Oosterdoksstraat 114, 1011DK Amsterdam, +31207575495, lauri.koponen@tomtom.com

Abstract
An advanced map matching algorithm is proposed, which embeds the road topology inherently in a set of path candidates. Paths are treated and evaluated independently using multiple criteria, with paths being ranked and a best path chosen, which makes the algorithm reliable and robust in different scenarios. The algorithm is efficient in both accuracy and computational speed, making it suitable to use in the mass-market navigation devices throughout all road transport environments.

Keywords: map matching, road connectivity, path candidates

Introduction

The horizontal positioning accuracy of current mass-market GNSS receivers is from approximately 5 m up to 100 m, and is highly influenced by the environments that can block signals and/or introduce multi-path errors. Digital maps contain errors due to digitization to polylines and representation of roads by centerlines. As a result, it is difficult to ensure that the direct footprint of the positioning data can be registered properly on a digital map. Map matching has thus been regarded as an indispensable process to correlate the digital map with the positioning records (including e.g. GNSS absolute positioning data, and/or dead reckoning (DR) relative positioning data) [1]. The matched location on the digital map is then used to topologically correct the travel route and associated road attributes and statistics shown to the driver [2].

The earliest map matching approaches include point-to-point and point-to-curve matching [3,4], that snap individual recorded positions to the nearest road nodes or segments, based on a distance measurement. However, due to the dual uncertainty and inaccuracy involved in both the positioning records and the map road network, these approaches are unreliable and error-prone. Urban areas with high road density, complex intersections, and highways with parallel or stacked roads are especially difficult to solve correctly. Improvements have been suggested in two trends. One trend is to introduce more measures in addition to the distance measurement. Heading is used the most [4,7]. The average distance travelled on the candidate road link is considered in [8]. Road attributes, such as turn restrictions, are also regarded as useful information to filter unqualified road links [10]. Another trend of improving map matching quality is to incorporate road network topology to maintain the map matched results topological integrity. In other words, road candidates should only contain those roads that are reachable from the previously matched locations. The topology relations, especially road connectivity, have been used to either eliminate unreachable candidates [11] or to increase the hypothesis of certain candidates at ambiguous intersections [5,12,13]. The performance is sensitive to the selection of the initial match; a transient matching error can lead to successive mismatches. Additionally, once there are interruptions in the positioning data stream, such as those caused by tunnels, tree canopy, or a deep urban canyon, the road connectivity is lost and requires a fresh initialization [11]. This paper utilizes and extends both trends: (1) it makes full use of input information by equally fusing them to a single likelihood by belief fusion [1], allowing easily merging additional data in the future, such as route data and camera image analysis results; (2) the road topology is inherently embedded in the candidate pool, assuring that close, but unconnected paths are excluded naturally. Path candidates are treated independently and completely define the working set of the map matcher. Strategies on how to expand and prune paths are also introduced, with special treatments to maintain the road connectivity during input interruptions.

Instead of using the point-to-point or point-to-curve matching, this paper utilizes curve-to-curve matching by connecting multiple positioning points to form piece-wise linear curves and match them against the road network curves. Curve-to-curve matching [2,4,5] is more precise since it eliminates the possibility of mismatch by a single input outlier, but it requires more expensive computation and its accuracy is limited by the dual positioning and map errors. In GNSS-denied environments like tunnels, either positioning inputs are not available at all or one can only rely on the shifted dead reckoning (DR) relative positioning [6]. In these situations, it is difficult to match the drifted DR trace to the correct road even using curve-to-curve matching. The DR drift can add up to hundreds of meters in just a few minutes. Some solutions combine map matching and positioning in a feedback loop to correct the drift incrementally [6,14], which in turn influences the following matches. However, a wrong map matched location in such coupling will consequentially lead to wrong drift correction and eventually a matching blunder. This paper proposes a unique path propagation step to isolate the shift from the normal error, allowing the curve-to-curve matching to work properly. It is a crucial procedure to offer reliable matching for a low quality map and in a GNSS-denied environment. The map matching algorithm is designed to be scalable and modular. In addition to the path propagation module, other important modules in the algorithm include multi-criteria scoring and ranking, and eventually choosing the best path. The algorithm is efficient in both accuracy and computational speed, offering flexibility in all transport environments.

Overall Architecture

The map matching algorithm is designed to be scalable and modular. It utilizes the curve-to-curve matching concept from the history of positioning trajectory to all possible different paths. Candidate paths are defined as all possible ways to traverse the map, passing from one node to another or one road segment to another, and as such, are ordered sequences of road segments.

Figure 1 illustrates the overall map matching architecture, which consists of two separate layers: the input processing and the map matching algorithm. The input processing layer is responsible for accepting positioning input, sanitizing it, and detecting special driving conditions such as big input gaps, driving in reverse, U-turns and tunnels. The map matching algorithm layer is responsible for constructing the path candidate pool, evaluating the match of the input trajectory against the candidates, and maintaining the path matching results by logically ranking paths and minimizing jumps between paths. Special driving conditions are handled separately at the beginning of the map matching algorithm. When a special maneuver is detected, the path candidate pool is analysed and prepared by properly constructing and expanding candidates. The rest of the algorithm, starting from the path propagation, never processes special maneuvers.

Path propagation optimizes the offset between the input trace and each individual path. It borrows the concept of “propagation” from the Kalman filter: the map matcher updates the best position along each path epoch by epoch to maintain the best shape similarity while compensating for offset. Due to the offset removal, the followed matching steps will be valid even with a big input drift as in the DR tunnel cases. Position matcher and heading matcher attempt to find the best possible match based on their own measurement, and may therefore end up propagating to different positions along the path. A certain heuristic is thus applied afterwards to select a single best position on each path, as the expected output of the algorithm is a single map-matched position. After that, paths are scored by different matchers, including the position matcher, the heading matcher, the distance travelled matcher and the speed limit matcher. More matchers can be considered if more input information is available. Scores are obtained based on curve-to-curve matching. Multi-matcher scores are combined to produce a single likelihood for each path. Then, paths are ranked for the best path selection. Due to the dual uncertainty and inaccuracy in both the positioning records and the map data,

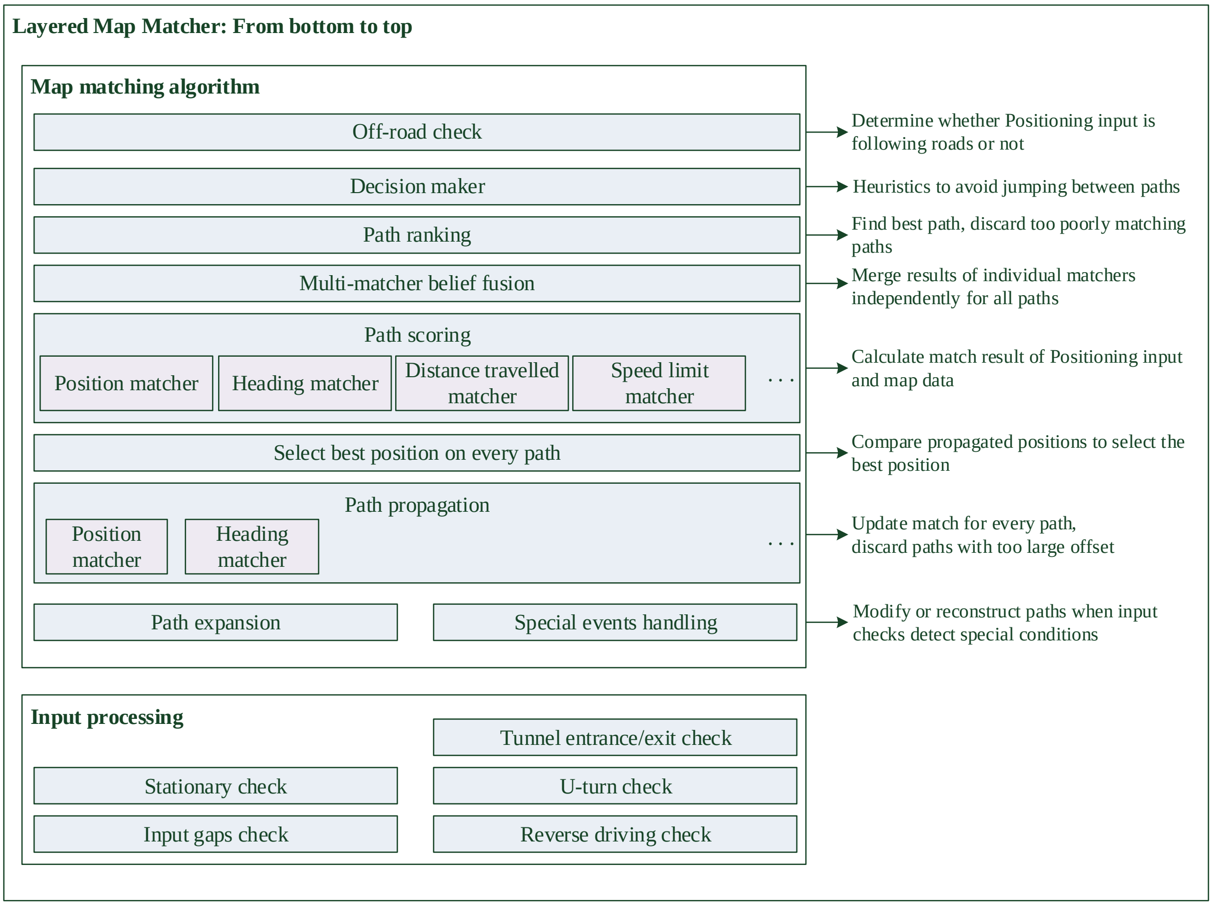

Figure 1 – Map matching overall architecture

the best ranking path might ultimately turn out to be incorrect. A decision maker is designed to minimize jumps between paths. Finally, a separate point-to-curve method is implemented as a last resort to find new matching candidates if none of the paths are found to be close to the input trace.

Path Candidate Pool Based on Road Topology

Path expansion

Path candidates are initialized as individual road segments, which are close enough to the first positioning input sample in both distance and heading, bound by input accuracy in an enclosing rectangle. As new positioning data samples are received, paths are expanded forward from the path heads to connected road links. Whenever an intersection is reached, the path candidate is copied so that the different expansions with the newly added road segments can be accommodated. The copied paths are treated independently by the rest of the algorithm. The new segments can be called “child segments,” while the previous segments before the branching point are the common ancestors of all the branching paths.

To avoid the number of path candidates growing exponentially, the path expansion for each candidate is only triggered when the previously best matched location on that path is close to the path head and the new input can be potentially matched beyond the path. Paths are never expanded backward, along the same road segments from which they arrive. Paths are also not expanded against traffic flow to clearly forbidden roads, such as highways. Common driving habits require path expansion against the direction of traffic on normal one-way roads.

Candidate management

Candidate management is necessary to reduce the path candidate pool size while preserving well-matching candidates. Reducing the pool size includes two steps: aggressively pruning the unused path tail on each path to make the path shorter as the positioning input progresses; and discarding incorrect paths from the candidate pool. Paths are tracked independently. A less likely path might ultimately turn out to be the correct one. Discarding of incorrect paths should therefore be conservative to avoid mistakes. The algorithm evaluates the following properties to discard paths. The first one is the offset from the positioning input to the matched location. If this distance significantly exceeds the input horizontal accuracy, the path is discarded. The accuracy must be used conservatively; otherwise the drift caused by a DR positioning solution may cause paths to be mistakenly discarded. Secondly, paths with bad score ratios relative to the best score are discarded. The algorithm uses the score ratio instead of the score itself since absolute scores vary in different scenarios. For example, bad scores appear in deep urban canyons with GNSS signal blockage, while good scores occur on open highways. Thirdly, it is crucial to keep all paths unique, in the sense that any two paths, looking backwards for a certain distance, do not overlap exactly. Paths that overlap with a higher ranking path will be discarded. Overlap situations usually occur when paths are expanded through a branch, and later merge in a following intersection.

The candidate pool size must be kept within a feasible computational region for real-time mass-market devices. Figure 2 gives an example of the path candidate pool size changes when a car drives in a dense urban canyon in Amsterdam. As shown, the algorithm has great control over the candidate size thanks to proper candidate management strategy and proper path expansion.

Maintain road connectivity in special situations

U-turns are processed early in the map matching process when the input trace is detected to be a U-turn shape. New paths with opposite directions are added to the candidate pool in addition to the original paths. This ensures that the map matching can work continuously in special road shapes such as hairpin turns or roundabouts, or if a special input trace has been mistaken for a U-turn. Map matching treats both groups of candidates equally. Poorly matching paths will be found by scoring and ranking, and eventually discarded.

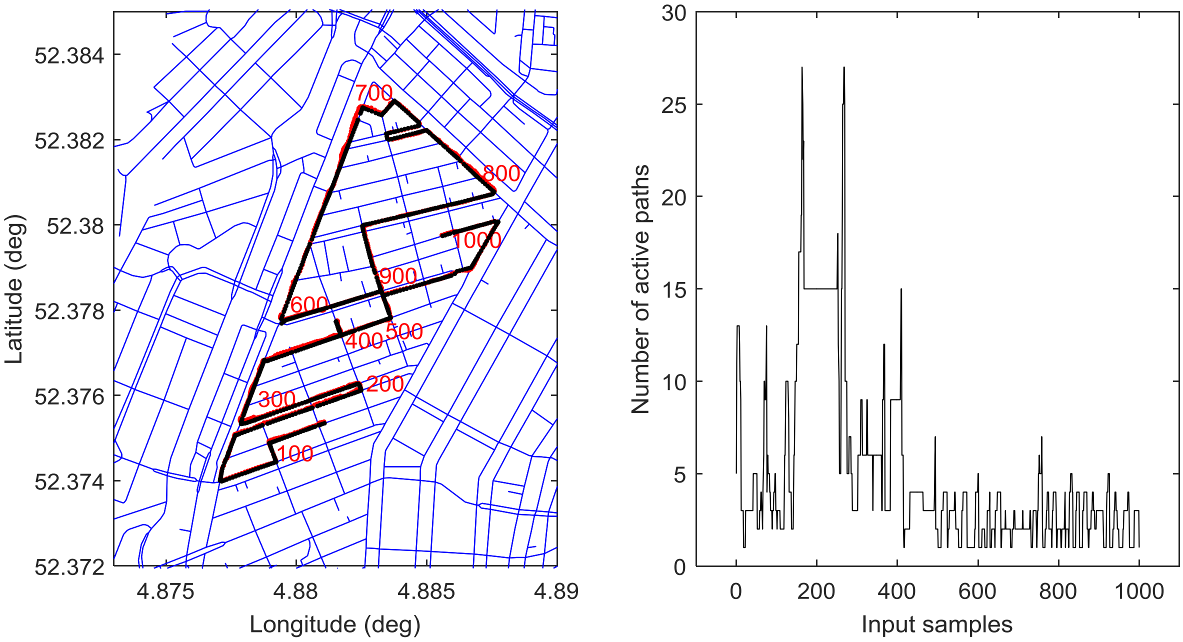

Figure 2 – Map matching in Amsterdam urban canyon. (a) Map matching results: red and black dots represent positioning trace and map matched locations, respectively. (b) Number of path candidates along the trip

Tunnels are another special situation, since GNSS signals are generally completely blocked. The question is how to maintain the road connectivity until the GNSS fixes are reacquired? Whenever there is a GNSS outage, it is possible to follow the roads marked as tunnels in the map, and expand only the paths that reach tunnel exits. In this way, the path expansion is correctly prepared for the first reacquired GNSS fix, which can then be captured at the tunnel exit and continually matched to the path. Figure 3 illustrates this situation. Paths are expanded towards all tunnel exits during the GNSS outage, while waiting for the first reacquired GNSS fix to be confirmed.

Path Propagation

During the map matching update, each matcher on each candidate path is first propagated forward with the new positioning input, producing an estimate of the correct place along the paths, where the input is expected to match.

Because both the digital map and the input trace might be shifted, it is impossible to always obtain a correct curve-to-curve match without compensating for horizontal position shift. For the position matcher, the best matched position along the path should be defined as the position that minimizes the curve dissimilarity after enumerating all possible shifts. However, the curve dissimilarity, represented by score, is nonlinear and expensive to optimize. A compromise is thus made in the map matching algorithm to propagate the shift in the path

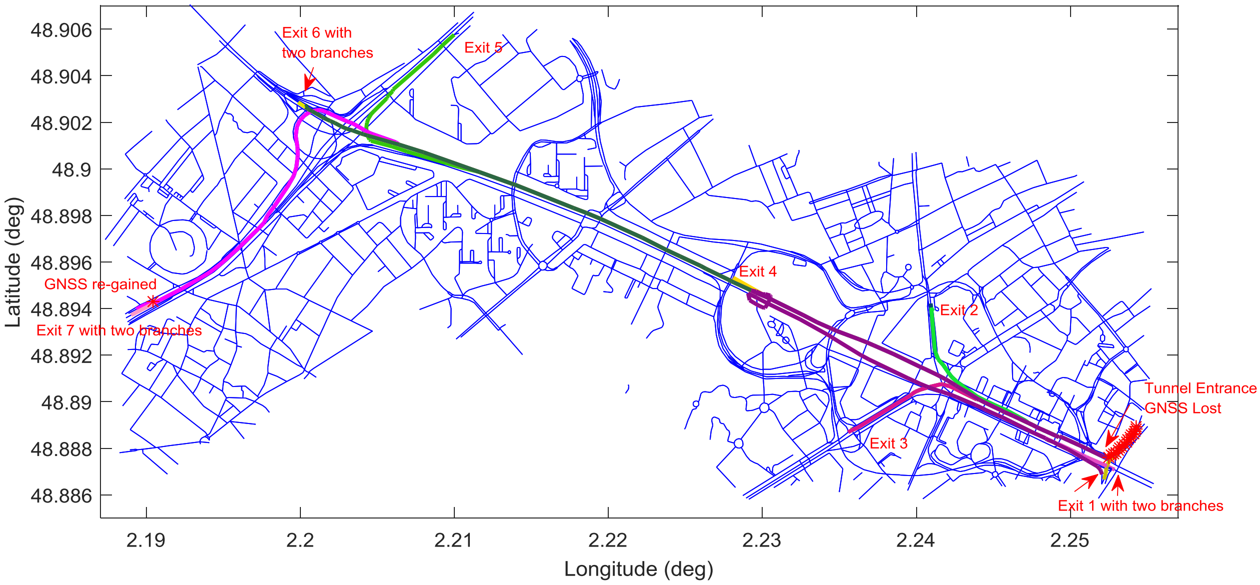

Figure 3 – Tunnel path expansion during the GNSS outage. Paths are represented by different colours, GNSS trace with red stars (bottom right).

propagation module before scoring. The initial shift is simply assumed to be the cross-track offset between the first input sample and the path curve. As inputs progress, the input trace is adjusted by the previous offset before calculating the new cross-track error, which is then used to update a new offset. Therefore, the offset is accumulated epoch by epoch, which is computationally efficient. It is worth to mention that a cross-track offset before a turn becomes an along-track error after the turn. In order to compensate the along-track error, offset optimization is proposed. It backward propagates the translated input trace along the path and (weighted) averages the obtained variety of offsets for an offset correction. Offset optimization guarantees a better shape-based match to the path, but it is relatively expensive and is therefore only executed under certain conditions, e.g. after turns.

Like the position matcher, the heading matcher also processes two-dimensional data, consisting of accumulated travelled distance and heading. The road geometry is represented by the accumulated path length and the road segment heading in the heading matcher. The input trace is represented by the input distance travelled and the input heading. The curve-to-curve matching then measures the discrepancy of these two (stair) step-like turning functions. Notice that the accumulated distance along the path often differs from the true distance driven, due to the simplified map and/or the absence of road width information. For example, a corner that is made up of several successive road segments is usually longer than the actual distance that a driver will travel, while changing lanes along a wide straight road results in a longer input distance travelled than the road segment accumulation. This can be compensated in the heading matcher path propagation. Offset optimization is also implemented to minimize the discrepancy of two turning functions, and also to prevent the heading matcher from skipping over problematic turns along the path when the distance travelled increases.

Path propagation not only optimizes the offset, but also produces an estimate for the best matched location. Some matchers, such as a speed limit matcher and a distance travelled matcher, do not have enough information to estimate a best matching position from their inputs and may skip the path propagation step and rely on the resulting matched location from other matchers.

Path propagation is essential to assure valid matching with a big drift in GNSS-denied environments. For example, Figure 4 depicts a tunnel case where, despite the DR input trace having drifted by more than 500 meters near to the tunnel exit, it is still well matched to the correct path both before and after the exit.

Path Scoring and Multi-Matcher Fusion

The position matcher utilizes the summed norm as the score, while the heading matcher treats the area of the summed-up rectangles between two tuning functions as score. They are relatively simple ways to quantitatively measure the curve-to-curve mismatch. Another matcher could be added for altitude changes. Other matchers such as the distance travelled matcher and the speed-limit matcher are based on the resultant best matched location after the path propagation to further confirm or deny the result on each path. As more measures are

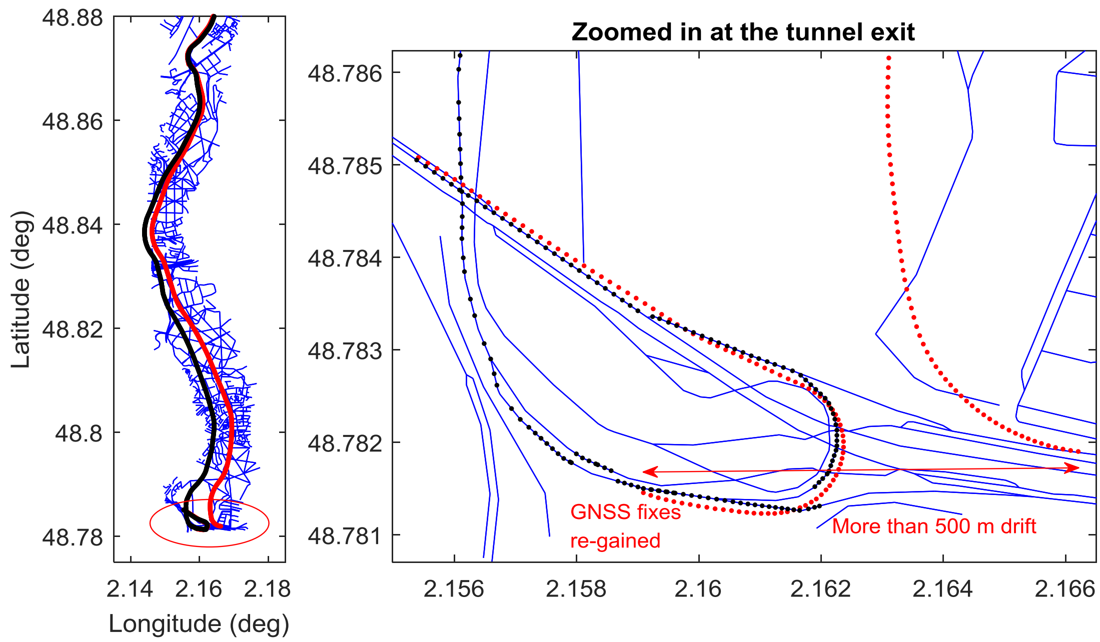

Figure 4 – Map matching from the drifted DR positioning trace to the tunnel path. Red and black dots represent the input trace and the matched locations, respectively

involved in the matching process, multi-criteria fusion is used in the algorithm, independently for each path, to distinguish the conflict between different criteria and to estimate the likelihood that the path is the correct one. Belief theory, also called Dempster-Shafer theory [7,9], is chosen for the multi-criteria fusion over other probability theories such as hidden Markov chain [10], fuzzy logic [8], and Kalman filter [9] because it can work even if the source of the belief values is not completely defined. Additionally, the Dempster combination rule is purely a cumulative operation, which makes it easy to expand and computationally inexpensive.

Path Ranking and Decision Maker

Path ranking and decision maker modules select the best matching path among all path candidates. Due to the separation of the path propagation and the path scoring, the information used to select the best path includes the offset between two curved traces and the belief likelihood that the path is the correct one. They are two different matching indicators. The offset is the vector to shift the input trace and therefore represents the input or map bias, while the likelihood is a quantitative index between 0 and 1 to represent curve-to-curve shape mismatch. Neither of them is truly reliable. For example, likelihood does not distinguish two parallel roads as they have the same geometric shape and equally match the input trace. In this case, offset will play an important role. On the other hand, offset cannot solely judge the match. In the DR tunnel case, the offset will continually increase due to the DR drift and the algorithm has to rely on the shape-based likelihood.

Path ranking merges these two matching indicators in a layered design. First, it classifies all path candidates into two groups based on the offset, and then continues to sub-classify the group with a better offset into sub-groups based on the likelihood. Within each group, paths are further ranked in the descending order by the likelihood. Ranking produces the best matched path for the current input. However, it does not prevent jumping between paths when the best matched path varies from one sample to the next. The decision maker is then used to minimize jumps; it compares the current best ranked path against the previous best for a decision either to jump to the new best or to continue using the previous one.

Conclusions

This paper introduces an advanced topological map matching algorithm. It fully utilizes the road topology and restrictions to construct and maintain a pool of possible path candidates, which are used to match against the positioning input trace. The algorithm is scalable and designed to be modular, consisting of path expansion, path propagation, best position on path selection, path scoring, multi-matcher fusion, path ranking, and decision maker modules. Special situations such as U-turns, tunnels, and driving in reverse are handled separately to prepare correct path candidates for the map matching algorithm. Since topological integrity is maintained inherently in the path candidate pool, solving many difficult situations such as parallel roads and roundabouts becomes a positive side-effect.

The map matching algorithm in this paper treats partial curve-to-curve matching discrepancy as either the map error due to road centerline simplification or the positioning input bias due to drift. These discrepancies are further eliminated from the path scoring for the best path selection, resulting in a better shape matching. The algorithm also uses full input information such as position, heading, distance travelled and speed limit as multiple criteria in path scoring, leading to more reliable performance and lower latency. The algorithm has a good balance between matching accuracy and computational speed, and is suitable to use in mass-market navigation devices.

References

1. Quddus, M.A., W.Y. Ochieng, R.B. Noland (2007). *Current map-matching algorithms for transport applications: state-of-the art and future research directions*. Transportation Research Part C: Emerging Technologies, vol.15, no.5, pp. 312-328.
2. Zhou, J., R. Golledge (2006). *A three-step general map matching method in the GIS environment: travel/transportation study perspective.* International Journal of Geographical Information System.
3. Bernstein, D., A. Kornhauser (1998). *Map matching for personal navigation assistants*. The 77th annual meeting, The Transport Research Board, 11-15 Jan 1998, Washington, D.C.
4. White C.E., D. Bernstein, A.L. Kornhauser (2000). *Some map matching algorithms for personal navigation assistants*. Transportation Research Part C, vol. 8, pp. 91-108.
5. Xi, L., Q. Liu, M. Li, Z. Liu (2007). Map Matching Algorithm and Its Application. ISKE-2007 Proceedings, part of series in Advances in Intelligent Systems Research
6. Aggarwal, P., D. Thomas, L. Ojeda, J. Borenstein (2011). *Map matching and heuristic elimination of gyro drift for personal navigation systems in GPS-denied conditions*. Measurement Science and Technology, vol. 22
7. El Najjar, M.E., Ph. Bonnifait (2002). *Multi-criteria fusion for the selection of roads of an accurate map*. 15th Triennial World Congress, Barcelona, Spain, 21-27 July 02.
8. Syed, S., M.E. Cannon (2004), Fuzzy logic based-map matching algorithms for vehicle navigation system in urban canyons. ION National Technical Meeting, 26-28 Jan 2004, San Diego, CA.

9. El Najjar, M.E., Ph. Bonnifait (2003). *A roadmap matching method for precise vehicle localization using belief theory and Kalman filtering*. In proceedings of ICAR. The 11th International Conference on Advanced Robotics, June 30-July 3, Coimbra, Portugal, pp. 1677-1682

10. Goh, C.Y., J. Dauwels, N. Mitrovid, et al. (2012) Online map-matching based on Hidden Markov model for real-time traffic sensing applications. 15th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 776–781.

11. Marchal, F., J. Hackney, K.W. Axhausen (2004). Efficient map-matching of large GPS data sets - Tests on a speed monitoring experiment in Zurich. *Arbeitsbericht Verkehrs- und Raumplanung*, Institut fur Verkehrsplanung und Transportsysteme, ETH Zurich, Zurich

12. Velaga, N.R., M.A. Quddus, A.L. Bristow (2009). Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems. Transportation Research Part C, vol. 17, pp. 672-683

13. Pyo, J.S., D.H. Shin, T.K. Sung (2001), Development of a map matching method using the multiple hypothesis technique. In Proceedings of IEEE Intelligent Transportation Systems Conference, 25-29 August, 2001, Oakland, CA, U.S., pp. 23-27.

14. Yu, M., Z. Li, Y. Chen, W. Chen (2006). Improving Integrity and Reliability of Map Matching Techniques, Journal of Global Positioning Systems, vol. 5, No. 1-2, pp. 40-46