Automated Construction of Road Networks from GPS Tracks

Weiping Yang
Analysis and Geoprocessing, Esri, Inc.
380 New York St. Redlands, USA
email: wyang@esri.com

***Abstract*--This paper describes a framework for automating road networks using GPS (Global Positioning Systems) track measurements. Through observation and experiments on the data, it is decided that automating road networks is done in a two-step process. The first step is to identify intersections of roads, following an intersection model that also identifies and holds tracking groups leading to the first legs of incident roads. The second step, road segments incident to intersection nodes will be iteratively discovered by moving probe lines perpendicular to the heading directions of the generated roads. Initial intersections are assessed through analysing turns of vehicle trajectories and characteristics pertinent to where roads meet. Statistical techniques are used on tracks in relation to probe lines to exclude outliers and to locate median positions as vertices of roads. The method described in this paper exploits topological and geometric measures about neighbourhood of roads and applies machine learning techniques that iteratively compute optimized results for these measures.**

Keywords – vehicle GPS data tracking; automated road extraction; road network analysis; geospatial data mining and knowledge discovery; machine learning.

I. INTRODUCTION

The availability of ever increasing amount of GPS data has given rise to the needs for capabilities of processing large quantities of data and of discovering knowledge, patterns, or actionable information. One practical need is to find roads from GPS tracks representing trip trajectories of vehicles moving on roads or parking areas. The roads can serve as base maps for mapmaking, trip planning, guiding navigations, or as updates to existing map databases. Compared to traditional methods of collecting transportation data through field surveying or remotely sensed images, GPS tracks provide an inexpensive, significantly massive, and timely data sources for conventional and emerging applications requiring road networks. This paper proposes an algorithm that generates roads from GPS tracks which are chronological records, for example, of Uber vehicle trips. No prior knowledge of existing road databases is assumed. Fig. 1 illustrates a study area which contains a dataset of Uber GPS points captured around San Francisco area (left). The tracks formed by chronologically connecting GPS points belonging to same vehicle trips are shown at the right.

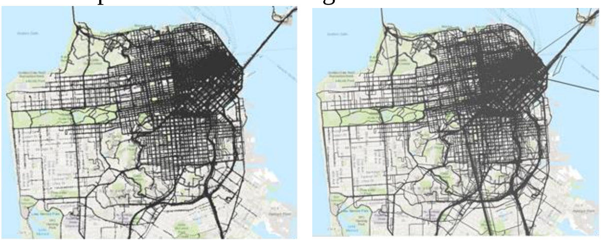


Figure 1. Uber GPS points and tracks.

As evident from the above figures, GPS points or connected tracks form clusters largely in linear shapes along roads in background images. Comparing the left and right maps in Fig. 1, one can see that “discrepancies” of data become a norm as arbitrary lines can be observed crossing the map. These lines are indeed caused by errors in GPS data. In addition, there are GPS tracks or sections of which that appear ambiguous on which roads they are supposed to adhere to. These ambiguities are noises among the largely clustered data. Furthermore, in the highly built-up downtown area where GPS location estimation becomes widely inaccurate, large number of spurious points have severely blurred street patterns. The errors and noises in GPS data collection presents additional challenges for devising a robust automated method of extracting roads.

Problems of extracting roads from GPS tracks have been tackled ever since GPS became a popular addition to vehicles. The diversity of published methods reflects usages of the extracted road structures. One type of the objectives, refining and enriching existing maps for advanced trip planning and navigation, requires accurate road geometries, better connectivity, multi-lanes, and intersection structures [1][2]. The algorithms to this end usually depend on existing road networks; and have a prerequisite for a map matching [3][4] algorithm to find correspondence between GPS tracks and existing roads. Another type of goals is more general. It does not require the existence of road maps but attempts to extract roads from scratch on GPS tracks only [5]-[9]. This type of algorithms usually applies statistical and machine learning techniques, such as least squares, k-means or density-based spatial clustering of applications with noise (DBSCAN), to discover road network patterns. The result roads can be served as base maps for new development areas or as timely updates to existing databases. Biagioni and Eriksson [10] made a comprehensive survey on earlier methods of map generation and pointed out the issue of lacking automated procedures for verifying and evaluating results. Ahmed et al [11] followed up with a book summarizing the published major algorithms on map construction and highlighted three types of algorithms, namely point clustering, incremental track insertion, and intersection linking.

The method proposed in this paper constructs road networks, as planar graphs, by discovering linear and connectivity patterns from GPS tracking points. Upon observations and experiments on data, it is decided that the first step is to identify intersections of roads, based on an intersection model. In the second step, road segments incident to each intersection node will be iteratively discovered by progressively moving probe lines perpendicular to the heading directions of the trailing road segments. Statistical techniques are used on tracks in relation to probe lines to exclude outlier tracks and to locate median positions as vertices of roads. The output is a dataset of road features with an average speed and a count indicating the number of vehicles travelling on each road. The method that determines intersection first is in line with the approach taken by Fathi and Krumm [12]. Unlike training a shape descriptor and time-exhaustively moving it around to detect intersections [12], the intersections in this paper are discovered by evaluating and clustering turns of trajectories, so the intersections are found analytically and are more likely corresponding to real world road junctions with stop signs. Furthermore, the validation process designed in this paper, in addition to determining final positions of intersections, accomplishes a discovery of similar tracks belonging to same roads incident to intersection nodes.

Testing and evaluating the goodness of generated road networks faces a challenge to producing automated qualitative and quantitative assessment. Fortunately, recent development in feature matching [13] and the commercially available Detect Feature Changes ArcGIS® geoprocessing tool [14] can provide comprehensive comparisons with existing roads in databases.

Section II will be devoted to identifying intersections. It is followed by Section III, analyzing track orientations bearing on which incident roads starts or ends. In Section IV, road segments, starting from an intersection will be extracted in a progressive fashion. Preliminary results of road networks will be presented in Section V and evaluated in Section VI. Discussion and future work will conclude the paper.

II. IDENTIFY INTERSECTIONS

In a prime [15] planar graph, intersections are nodes at which road segments meet as edges. Real world roads projected on a plane can be viewed as crossing each other, not all of the crossing points are intersections, i.e., there are no stop signs or roads not crossing at the same elevation level. Computing all intersections between two tracks is not only expensive, as there are too many of them, but also inconclusive. The results have to be screened considering elevations and other factors. On the other hand, real world intersections can be identified through a number of statistically significant indications. For example, 1) vehicles must stop at the stop signs or red lights; 2) vehicles are able to turn left or right; 3) the degrees of turning angles formed by adjacent roads cannot be arbitrary; and 4) the number of incident roads unlikely exceeds 6. Considering these indications and performance effectiveness, it is decided in this research that road intersections will be sought after first. Apparently, the first indication is of temporal: it can be revealed by observing longer time laps between two consecutive points around an intersection. This indication can help determining intersections with no turning tracks. Establishing a reliable tolerance for the laps, however, needs to be further investigated. The other indications can be captured through metric measures. In this paper, intersections are primarily identified by traffic turns.

A. *Determine Turns at Intersections*

Turns in a trajectory, from one road to another, can be captured by turning angles. It is assumed most roads intersect by an angle near 90 degrees. If two legs of a trajectory before and after a turn form an angle, say 90 ± δ, where δ is a tolerance threshold, a turn point could be located. In the experiment taken by this research, we calculate turn points considering three turning cases shown in Fig. 2.

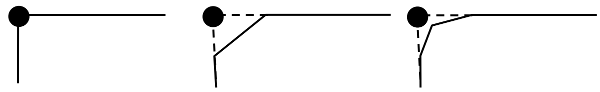
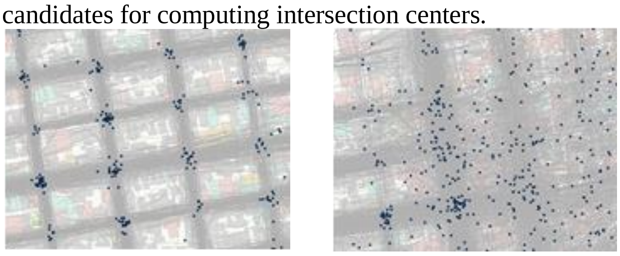


Figure 2. Three cases determining turns.

Additional screening processes are needed, however, to disqualify turns in parking lots, which are characterized by multiple turns within an area with short legs. It is desirable to identify the trajectory sections roaming around parking lots, and to exclude these GPS points from participating in extraction of roads. The accumulation of these GPS points may help to outline parking areas. Observation shows most of the parking lot GPS points are occurred at the beginning or end of trips.

B. *Find Clusters of Turns*

Intuitively, turns at the same intersections should be located near a real intersection center, which form a cluster. Finding these clusters, using a DBSCAN, involves building a spatial index to facilitate searches and expanding neighboring turns from any seeds. The prime criteria for stopping expanding a cluster is the distance between any two neighboring turn points. Additional criteria may consider the shapes of clusters which, ideally, are round and limited in sizes. Compact clusters, round and gathered with large numbers of turns, are excellent candidates for computing intersection centers.



a. Turn clusters in grid-like street area b. Turn clusters in urban core area

Figure 3. Clusters of turn points.

Fig. 3a illustrates a distribution of turn points (black dots) within a regular grid street area. Referred to the background map image, most intersections are superimposed with densely clustered turn points. At the low traffic volume areas, only one or two turns can be identified near intersections. The clusters with one or two turn point will also be considered for the reason that we don't wish to miss an intersection. This needed relaxation brings in a lot of dubious clusters. As shown in Fig. 3b, there is a large number of loosely distributed turn points in the urban core area where GPS points are scattered, mostly due to the street canyon effect. Verification is a must to exclude false clusters.

III. VALIDATE CLUSTERS AND COMPUTE INTERSECTIONS

Turn point clusters are intended for computing centers of road intersections. Validating turn clusters will largely rely on this purpose, by examining all nearby tracks passing through or turning at a cluster center. This process will also identify and group tracks that are statistically appropriate to form coherent clusters for roads which are, regardless outgoing or incoming, incident to the cluster center, i.e., the intersect. Fig. 4 illustrates turns that might or might not lead to intersections.

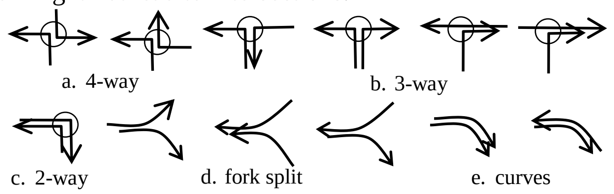


Figure 4. Intersection model.

When trajectories of multiple tracks form one of the configurations in cases a, b, and c, an intersection will be validated. Cases in d will not be assessed as intersections at this time but will be recognized as fork-like splits later in extracting roads. Fork-like splits will be dynamically treated as intersection nodes. The 2-way intersection in case c qualifies only when it constitutes a near 90-degree turn. Otherwise it is just like a curve as one in cases e.

A. *Collect Tracks Involved Around an Intersect*

Validating a cluster uses a square box centered at its mass center to clip all tracks intersecting the box. The size of the box, say 60x60 m^2, is experimented to cover the entire intersection area. The clipped lines, mostly straight some with a vertex within the box, will be served as the basis for the analysis (Fig. 5), as described below.

For each clipped straight or near straight line, a projection from the box center is made. If the foot is in the box, the line will be split into two oppositely directed lines, as shown in the left box. For a non-straight line, there must be a turning point and the turning angle is near 90 degrees. The turning point will be used to extend the two sections from both ends to meet the box border. The two thus formed straight lines will start from the extended border points, shown in the middle box. The extension will be needed later for computing intersections. As is shown in the right box, without the extension, (dashed parts), some tracks will be missed for intersecting and the initial mass center cannot be accurately located.

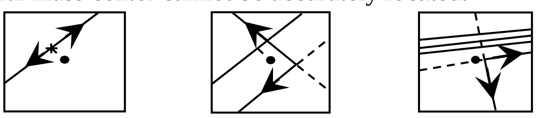


Figure 5. Clipping boxes for intersection analysis.

B. *Group Tracks by Orientation*

Orientations of straight lines obtained from above will be classified for grouping similar tracks. We use an 8-sector circle to classify track clips (Fig. 6), i.e., an 8-means clustering. The east axis is on degree 0 and angles increases anticlockwise. Each sector has a range of 45 degrees to hold orientations falling in. For example, the first group is the shaded sector. It holds orientations from -22.5 to 22.5 degrees, as (-22.5, 22.5]. Orientation values in the range of (22.5, 45] will fall into the second group, and so forth.

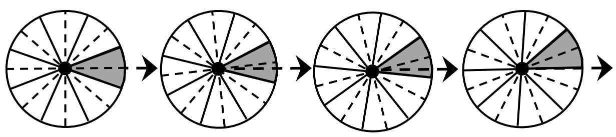


Figure 6. An 8-sector circle for clustering orientations.

After assigning clipped tracks into sectors based on their orientations, some groups may have large number of occurrence and some may end up empty. Empty groups will be removed. Mean values and squared errors of each group orientations are calculated. The template circle will then be rotated an interval of 7.5 degrees anticlockwise from its initial sector division, up to 3 times, which brings the first group ranges to be (-15.0, 30.0], (-7.5, 37.5], and (0, 45.0], respectively. After each rotation, clipped tracks will be re-assigned and the mean and squared errors re-calculated. The new error values will be compared with the previously saved ones. If they are not better, the rotation is stopped. Experiment shows that the optimized results are obtained after 1 or 2 iterations.

C. *Remove Unlikely Incident Tracks*

Track clusters with three or more groups will be further validated. False track groups, that would lead to plausible incident roads, need to be eliminated. Given the fact that we use an 8-means clustering method, at most 8 cluster groups may be initially produced. Fig. 7 shows 8 initially grouped tracks by orientation (left) and the final groups remained after removal of false groups (right). One distinguished characteristic about intersections is that most of the times, there is at least one pair of track groups (like Groups 1 and 6) would lead to through traffic roads before and after the intersection. Furthermore, straight-through tracks usually form larger groups. This observation is useful in eliminating smaller groups that are slant to through-pairs. The analysis and reasoning below illustrate the removal of plausible groups.

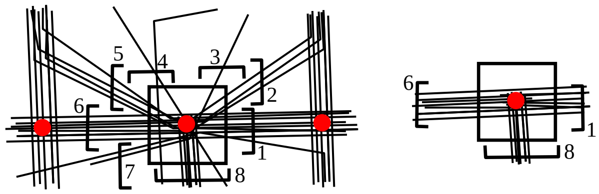


Figure 7. Eliminating false track groups.

Group 3 can be immediately removed due to lacking confidence for a determination of an incident road from a single track. Groups 2 and 5, slanting to a through-pair, are removed because they are likely the shortcut tracks missing a GPS point near the neighboring intersections. This hint can be verified by looking at the rightness of the sum of the turning angles at both ends of a slanting track segment. Group 4 is removed for it does not have an agreeable orientation. Group 7 will be removed because it slants to the through-pair with an angle smaller than 30 degrees. Of the remaining groups, the lowest track in Group 1 will be removed because its orientation is an outlier compared with the group mean. The left track in Group 8 is removed because its intersect position on the square box is an outlier. The right track in Group 8 is excluded because both its orientation and intersect position are outliers to respective means of the group.

The clustering that ends up with only two groups will not result in an intersection, if the difference of their mean orientations is not an anger near 90 degrees. Two track clusters forming a shallow angle are likely trajectories that traverse through a circular curve. They will not be considered further.

 ISBN: 978-1-61208-687-3

D. Adjust Intersection Centre with Validated Tracks

After removing plausible track groups, intersections are left with mostly 3-way and 4-way traffic routes. Some 5-way intersections exist, but 6-way intersections are rare the relationship of whose traffic routes usually need to be sorted out by additional analysis and reasoning. The final intersection point will be computed with the validated tracks. Fig. 8 shows the final intersections (red dots) on top of turn points (black dots).

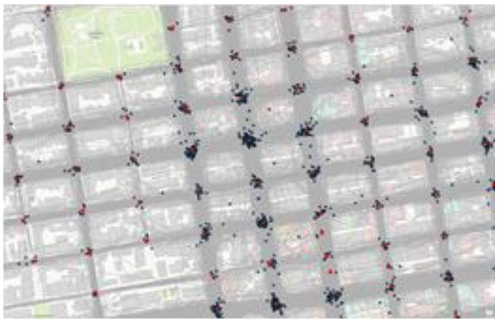


Figure 8. Verified intersections shown as red dots.

IV. EXTRACT ROADS

With a successful establishment of road intersections, extracting roads becomes relatively easier and controlled. Recall that associated with each intersection node are emitting orientations of incident roads which indicate the directions where the roads extend. All extraction will do is to use probe lines perpendicular to a road orientation and to progressively discover concentrations of tracks likely traversing on the road. Once a concentration is located, calculating and analyzing intersections of the tracks on the probe line, eliminating tracks unlikely traversing the road, and then taking the mean intersection point as the next vertex. New probe lines will be progressively moving ahead based on what has been discovered previously. A road will be terminated once another intersection node is discovered or no proper tracks can be found by the last probe line. This section describes what need to be considered in each step.

A. Sort Road Initial Segments by Track Frequencies

Firstly, the result from Section III.C, which are all intersection nodes and track groups organized together by similar orientations, will be sorted in descending order by the number of tracks in each group. Utilizing a priority queue holding the sorted track groups, each element of the queue contains the node ID, intersection point, the orientation to extend the first segment of a road, and the count of tracks in the group. The reason for earlier extracting roads from more heavily concentrated tracks is, the more tracks found traversing on a road, likely the more accurately the road can be extracted. The earlier accurate knowledge can be discovered, the easier successive analysis on insufficient data can be made. Since heavily traveled roads likely represent major roads or freeways, early extraction of them can help to control quality of a hierarchical road network.

B. Probe and Compute Road Vertices

Iteratively popping the top element from the priority queue, we will have the node location as the first vertex of a working road, the initial orientation bearing on which the first road segment will be proposed, and a list of track IDs passing the intersection. The initially proposed second vertex is a polar point whose coordinates are determined by an offset distance from the first vertex and the bearing orientation. To finally determine the first road segment, a probe line will be utilized. The probe line is centered at the polar point, has the length of a specified road width, and is perpendicular to the bearing orientation of the proposed road segment (Fig. 9). Intersections of the probe line with the known tracks associated with the node will be computed. From the intersections, the second vertex is obtained by an k-means clustering. The previous and the current vertices forms a road segment, which provides an updated orientation for the next probe line. Retained for a new probing are the track IDs and their intersection points with the previous probe line. Fig. 9 illustrates the terms of entities used in this section, and their relationships.

Unlike the initial probing where the tracks are already known to a road branch associated with the intersection node, spatial searches will be needed for the second probe line and onwards to find intersecting tracks. It is obvious that there could be tracks involved in previous probing will no longer be found (fading away) and tracks not seen previously be discovered (emerging). It is also easy to understand that not all intersected tracks should be used for computing a new vertex. Erroneous tracks should be identified and be eliminated. This research uses the following clues to identify erroneous tracks:

- Firstly, most of the intersected tracks should be known and are continuous from the previous probing. For each known track, computing its moving orientation from previously retained and current intersections, and comparing it with the extended orientation of the previously extracted road segment, if the orientation difference is too big, say greater than 60 degrees, the track will not be used.
- Secondly, newly found tracks will be more carefully inspected for use, considering their orientations and occurrence frequencies. If an emerging track does not satisfy orientation requirement but its intersection falls in the probe intersect point range, its occurrence frequency will be increased so it might be admitted for use in next probing.

This iterative process continues until an intersection node is found on the way, or a probing line catches no or just one track. In the case that an intersection is met, the road will be terminated there. In the other case, the road may end dangling, or additional analysis is needed for detecting possible turns that are missed in Section II, to be described next.

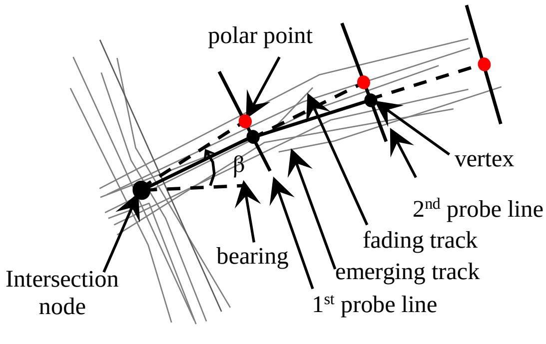
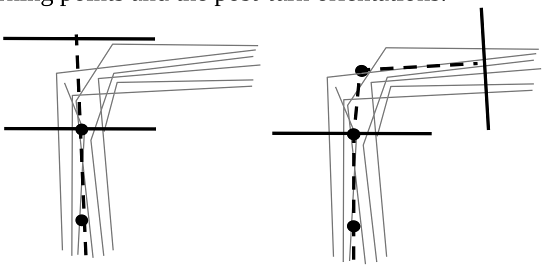


Figure 9. Terms and concept involved in probe lines.

 ISBN: 978-1-61208-687-3

C. Detect Turns

In the case that a probe line does not find any proper intersections, it is possible that the tracks have sharp turns, like what is shown in Fig. 10a. The task here is to find a turning point and an updated moving orientation after the turn, like what is shown in Fig. 10b. The turning point can be found by examining one or two vertices of a track near the last probe line intersection, and two segments prior to and after the vertices. If the angle formed by the two segments is near 90 degrees, a turn and changing orientation are computed. Doing so for all tracks retained in the last probing and taking the means of both the turning points and the post-turn orientations.



a. A new probe line intersects no tracks

b. Detect turning point and determine new orientation

Figure 10. Finding a turn and modifying probing orientation.

D. Reach an Intersection Node

Every time when a new polar point is obtained to start a probing, a search in the vicinity of the point will be made to check whether there is a proper intersection node to snap onto. For each node found, a verification process will be carried out to make sure the bearing orientation of the incoming road matches the orientation of a track group associated with the node. If no such a match can be found, a decision needs to be made whether to keep or discard the generated road. In the experiment of the paper, the number of tracks contributed to the last road segment will be the key factor for the decision. If the number is greater than 20, the road will snap onto the node; if less than 10, the road will be discarded; otherwise, the road will end and dangle at the last vertex. The verification has been proven especially useful in areas where GPS tracks are messed up to avoid erratically generated lines.

Once a snapping node is determined, the node ID and the orientation corresponding to the track group will be marked as processed, so the same road will be not be extracted again from the other end.

E. Split or Merge

As probe lines move forward, a fork-like split might be encountered (Fig. 11) when the range of intersections on the probe line becomes wider (left), or some of the track IDs suddenly missed from intersecting (right). Recall in Section III that fork-like shallow turns do not produce an intersection node, which will be created here. To find the split node, the probe line can be extended long enough to intersect the missing tracks. It is followed by finding two intersection clusters and their median positions on the probe line. With the knowledge of two track groups, a mean orientation for each group can be obtained. Two lines each passing a cluster center and bearing respective orientation can be used to find the intersection node. A final adjustment might be needed to make sure the node is on the heading direction of the last segment of the road extracted so far. After this, the current road will be terminated at the new node, which will be associated with the two splitting track groups and added to the priority queue of tracking groups.

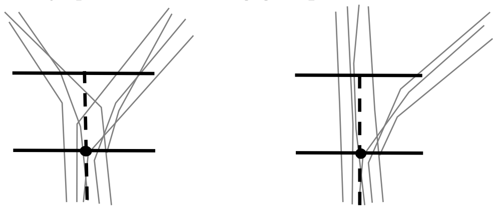


Figure 11. Detect splits by checking changes on probe.

Similarly, a merge node could be discovered when a range of intersection point on a probe line is shifted with new tracks appeared. The new IDs represent the other group of tracks involved in the merge. To find the intersection node, shallow turning vertices between the current and the previous probe lines can be found from which the merge intersection can be determined.

V. EXPERIMENT RESULT

Based on the algorithm, a geoprocessing tool to generate road networks from GPS tracking points has been prototyped using the ArcGIS® Pro platform. Applying the prototype tool, experiment on Uber GPS tracking data in San Francisco and surrounding suburb has been carried out. The dataset contains over 1 million GPS points covering urban core area with high rise buildings, streets of regular grids, and sparsely travelled country roads. It took about 30 minutes running on a desktop PC to create a road network with identified intersections. The result is displayed on top of a background image, shown in Fig. 12 where input tracks are in gray, the generated roads in purple, and the intersection nodes in red.

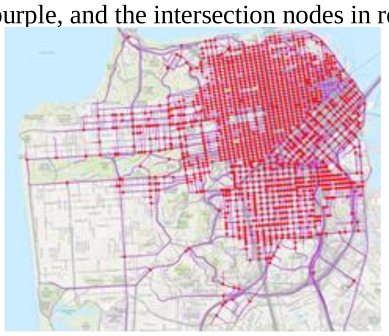


Figure 12. Road Network of SF area from GPS Points.

Fig. 13-15 illustrate zoomed-in images of the generated roads and intersections in grid-street, irregular road, and urban core areas.

Figure 13. Generated roads of regular grids.

Figure 14. Generated irregular roads.

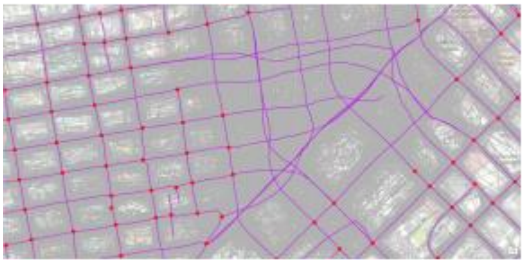


Figure 15. Generated roads around urban core.

It is easy to see that the proposed method works better when input tracks present clear linear threads and poor when there are no obvious patterns.

VI. THE EVALUATION METHOD

Evaluating the goodness of roads generated from GPS tracks, quantitatively, is challenging in that there has been no standard automated workflows to carry out the task. The challenge is aggravated due to the fact that tracking data is simply a snapshot of all possible travel patterns. Less travelled roads may not have any tracking records in the snapshot. In this paper, we explore a framework of evaluation and database updates by considering temporal aspects of GPS datasets. The commercially available GIS tool, Detect Feature Changes (DFC), in ArcGIS® Pro is utilized as a start for the framework. Based on a feature matching algorithm, the DFC tool takes in an existing road map as the base, a new road map as the update, a search distance and a change tolerance. The goal is: for each road in update dataset, find the correspondent road in base. If there is a match and the update road is within the tolerance buffer of the base road, the change type of the update will be NC – no change. If not within the buffer, or there is a 1:m / m:1 relationship, the change type would be S – spatial change. If an update finds no match, the type would be N – new. Any base roads with no matches in update would have the change type D – to delete. Let's use a map section (Fig. 16) to illustrate the evaluation process.

Figure 16. Existing streets (brown) on topographic image (gray).

Fig. 16 shows the existing road map in brown, on top of a topographical image (gray) for visual reference. The new roads (black) and the intersections (red dots) generated by the method presented in this paper are displayed in Fig. 17. Also displayed in Fig. 17 are GPS tracks (gray) and the brown base map for comparison.

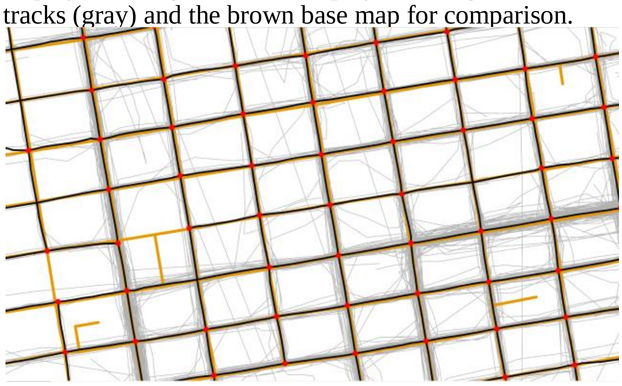


Figure 17. Automated roads and intersections.

Now running the DFC tool with the new and base, giving 10 meters and 5 meters for search distance and change tolerance, respectively. The result is a feature layer symbolized with change types (Fig. 18).

Figure 18. Symbolized DFC output layer.

The records holding the DFC output features are shown in Tab. 1. The table and the layer map provide interactive ways to inspect DFC result.

TABLE I. DFC OUTPUT, TOPOLOGICAL CHANGES HIGHLIGHTED

OBJECTID	SHAPE	UPDATE_FID	BASE_FID	CHANGE_TYPE	LEN_PCT	LEN_ABS	SHAPE_Length
135	Polyline	-1	33601	D	-1	-1	0.00085
136	Polyline	-1	33602	D	-1	-1	0.001653
74	Polyline	1115	31561	S	0	0	0.001912
76	Polyline	1153	33396	S	0	0	0.000999
80	Polyline	1212	31559	S	0	0	0.000888
83	Polyline	1338	31835	S	0	0	0.003342
87	Polyline	1445	32124	S	0	0	0.00191
96	Polyline	1601	33317	S	0	0	0.001655
105	Polyline	1842	31456	S	0	0	0.001641
99	Polyline	1718	32556	S	0.002827	0	0.000978
98	Polyline	1716	32947	S	0.002906	0	0.000904
101	Polyline	1751	32847	S	0.00291	0	0.001677
108	Polyline	2220	33155	S	0.030578	0.000001	0.00169
67	Polyline	495	32770	S	0.15369	-0.000001	0.00093
69	Polyline	582	32182	S	0.247986	-0.000004	0.001626

For example, one can highlight features with change type S and 0s under LEN_PCT, meaning 0 percent geometry change. the highlighted features will be displayed in the map (Fig. 19). By looking at the map result, it becomes obvious that these highlighted new roads match 2 or 3 features in base. The reason for the 1:m relationship is that there are no intersections identified with the tracks. At this time, the inspector could modify the change type to NC. Similarly, the inspector can examine the features with D type (brown roads in Fig. 19) and would find that there are no GPS tracks recorded on those roads during the time. Upon verification, they would not be counted as errors.

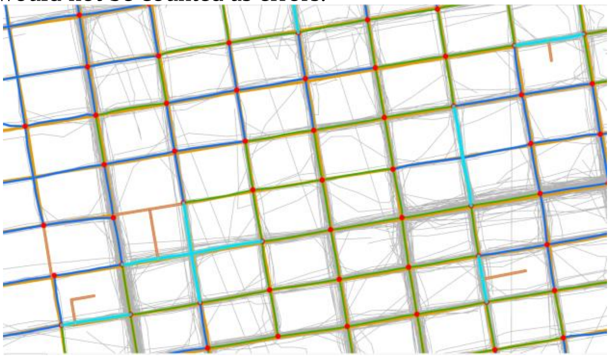


Figure 19. Features with topological changes.

After all suspected features are inspected, we can summarize the quality assessment for the map section: Of the 125 roads generated, all match corresponding ones in base. Furthermore, 71 roads are within 5 meters similar to their base peers, and all, except one, features with the S type are within 7 meters of their peers. The finding is encouraging for areas with clear patterns.

Other areas can be inspected based on the DFC result. It can be expected that there would be a lot of D type features that do not have roads generated from this snapshot. They might be available with additional tracking data taking from various other time periods. Automation could be enhanced to reduce manual inspection. For example, for each DFC feature with D change type, a proximity search can be made to verify that there are indeed no tracking data for the missing roads.

VII. CONCLUSION AND FUTURE WORK

The proposed method constructed road networks from scratch on input GPS tracking points by paying premier attention to analyzing road intersection nodes. The quality of nodes and roads produced is largely dependent on the quality of GPS tracks. The method identifies and excludes erroneous or outlier tracks by analyzing temporal gaps and excessive turns within a limited area, and by combining the use of spatial reasoning, statistical, and machine learning techniques.

Future work includes the following considerations:

- Paying attention to implementation details to learn and refine parameter values for better result;
- Adding algorithms to generate ramps, possibly with supervised ramp patterns and learning;
- Refining k-means clustering on the intersecting points along probe lines to identify and generate multiple lanes;
- Researching on quantitative and qualitative assessment methodology using feature matching tools and considering dynamic and real-time updates; and
- Developing a post processing tool to detect and if possible, to correct errors in the output road networks to prepare them ready for use in routing and network analysis.

ACKNOWLEDGMENTS

The Uber data used in this paper is obtained from: https://github.com/dima42/uber-gps-analysis/tree/master/gpsdata

The author appreciates the support from members in Esri Geoprocessing and GeoAnalytics teams.

This paper discussed an ongoing research. The author is solely responsible for any errors. The content should not be interpreted as any commitment by Esri to provide specific capabilities in future software releases.

REFERENCES

[1] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, “Mining GPS Traces for Map Refinement,” Data Mining and Knowledge Discovery, vol. 9, no. 1, July 2004, pp. 59-87.

[2] S. Rogers, P. Langley, and C. Wilson, “Mining GPS Data to Augment Road Models,” 5th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 1999, pp. 104-113.

[3] P. Newson and J. Krumm, “Hidden Markov Map Matching Through Noise and Sparseness,” 17th ACM SIGSPATIAL Int. Conf. Adv. in GIS, 2009, pp, 336–343.

[4] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk, “On Map-Matching Vehicle Tracking Data,” 31st VLDB Conf., Trondheim, Norway, 2005, pp. 853 – 864.

[5] S. Edelkamp and S. Schrödl, “Route Planning and Map Inference with Global Positioning Traces,” In Computer Science in Perspective (R. Klein, H.-W. Six, and L. Wegner, eds.), LNCS 2598, Springer, Berlin, 2003, pp. 128–151.

[6] S. Karagiorgou and D. Pfoser, “On vehicle tracking data-based road network generation,” ACM SIGSPATIAL GIS, 2012, pages 89–98.

[7] J. J. Davies, A.R. Beresford, and A. Hopper, “Scalable, Distributed, Real-Time Map Generation,” IEEE Pervasive Computing, vol. 5, no. 4, Oct. 2006, pp. 47-54.

[8] M. Ahmed and C. Wenk, “Constructing Street Networks from GPS Trajectories,” In: Epstein L., Ferragina P. (eds) Algorithms – ESA 2012. LNCS 7501. Springer, Berlin, Heidelberg.

[9] L. Cao and J. Krumm, “From GPS Traces to a Routable Road Map,” 17th ACM SIGSPATIAL Int. Conf. on Advances in GIS, 2009, pp. 3-12.

[10] J. Biagioni and J. Eriksson, “Inferring Road Maps from Global Positioning System Traces,” Transportation Research Record: J. Transportation Research Board, No. 2291, Washington, D.C., 2012, pp. 61–71.

[11] M. Ahmed, S. Karagiorgou, D. Pfoser, and C. Wenk, “Map Construction Algorithms,” Springer International Publishing Switzerland 2015, ISBN 978-3-319-25164-6.

[12] A. Fathi and J. Krumm, “Detecting Road Intersections from GPS Traces,” In S.I. Fabrikant et al. (Eds.): GIScience, LNCS 6292, 2010, pp. 56–69.

[13] W. Yang, D. Lee, and N. Ahmed, “Pattern Based Feature Matching for Geospatial Data Conflation,” GeoProcessing 2014: 6th Int. Conf. Advanced GIS, Applications, and Services, Barcelona, Spain. March 2014, pp. 64-69.

[14] D. Lee, W. Yang, and N. Ahmed, “Conflation in Geoprocessing Framework-Case Studies,” GeoProcessing 2014: 6th Int. Conf. Advanced GIS, Applications, and Services, Barcelona, Spain. March 2014, pp. 58-63.

[15] S. Porta, P. Crucitti, and V. Latora, “The Network Analysis of Urban Streets: A Primal Approach,” Environment and Planning B: Planning and Design, vol. 33, 2006, pp. 705-725.