Report 15-UT-033

NSTSCE
National Surface Transportation
Safety Center for Excellence

Matching GPS Records to Digital Map Data: Algorithm Overview and Application

Shane B. McLaughlin, Jonathan M. Hankey

Submitted: March 3, 2015

Technology
Impairment
Mobility
Infrastructure

Housed at the Virginia Tech Transportation Institute
3500 Transportation Research Plaza • Blacksburg, Virginia 24061

ACKNOWLEDGMENTS

The authors of this report would like to acknowledge the support of the stakeholders of the National Surface Transportation Safety Center for Excellence (NSTSCE): Tom Dingus from the Virginia Tech Transportation Institute, John Capp from General Motors Corporation, Lincoln Cobb from the Federal Highway Administration, Chris Hayes from Travelers Insurance, Martin Walker from the Federal Motor Carrier Safety Administration, and Cathy McGhee from the Virginia Department of Transportation and the Virginia Center for Transportation Innovation and Research.

ABSTRACT

Records of latitude and longitude pairs describing an approximate path of travel are logged by many types of Global Positioning System (GPS)-enabled devices. These logs of latitude/longitude (lat/lon) pairs specify geographic locations but include error inherent in the GPS system. Digital map data include representations of roads, trails, airways, etc., with links and nodes that are located geospatially but that also include error. The following report describes an algorithm for matching GPS points to the correct road link in digital map data by using road network connectivity. The algorithm was applied to the naturalistic driving data from the second Strategic Highway Research Program (SHRP 2), the largest naturalistic driving study to date. The data set from SHRP 2 consists of 5.5 million trips, which generated approximately 3.7 billion latitude/longitude pairs that needed to be matched to roads represented in digital maps. When identifying roads from GPS data at this scale, both the processing speed and the accuracy of the algorithm are important. To evaluate the output accuracy, a sample of 100 randomly selected trips was compared to a manual route identification. The results indicate that the algorithm assigned driving data to the correct link 91% of the time. When the driving data were not on a link, the algorithm correctly recognized this 86% of the time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

AADT	Average Annual Daily Traffic
GIS	Geographic Information System
GPS	Global Positioning System
VTTI	Virginia Tech Transportation Institute

CHAPTER 1. INTRODUCTION

BACKGROUND

Records of latitude and longitude pairs describing an approximate path of travel are logged or can be logged by Global Positioning System (GPS)-enabled devices, which have become commonplace in handheld devices, vehicles, and research applications. The latitude and longitude pairs generated by the GPS provide the location of the device on the earth’s surface at a given time. When viewed as a chronological sequence, these pairs also provide a record of the geographic path traveled by the GPS-enabled device. However, there are many applications where it is desirable to translate records of latitude/longitude pairs into records of the physical roads or paths that were traveled. For example, fleet-monitoring systems might want to identify roads on which a vehicle is traveling to compare the vehicle speed to posted speed limits. Connected vehicle technologies will likely benefit from identifying the specific road a vehicle is traveling rather than using general proximity information provided by latitude and longitude. Naturalistic driving studies, and other travel-related research methods, would benefit from being able to identify roads and then associate detailed information such as road class, pavement type, speed limits, etc., with the location.

The data sets developed for these types of applications can become quite large. For example, the second Strategic Highway Research Program (SHRP 2) Naturalistic Driving Study (NDS) generated approximately 3.7 billion latitude/longitude pairs. A digital map encompassing the roads on which the participants traveled includes over 40 million links. Matching such a large number of latitude/longitude pairs to the large number of potential roadway links requires an algorithm that is both efficient and accurate. This report describes an algorithm that was developed to address these challenges.

PREVIOUS WORK

Previous efforts to match GPS data with digital maps have employed common geospatial tools (Cannon, McLaughlin, & Hankey, 2009; Li, Gibbons, & Medina, 2014; Wu & McLaughlin, 2012). One approach is to define buffers around roads and then identify GPS points within those buffers. Another approach uses a nearest-neighbor computation or spatial join to assign GPS points to the closest road. A drawback of these methods, however, is that they frequently assign points falsely to an adjoining road if the GPS point happens to land near or on that road rather than on the road actually being traveled. Though it is possible to develop countermeasures to this type of error, such as described in Li et al. (2014), the solutions tend to be computationally intensive (e.g., a spatial join of many GPS points to many lines through space) and the tools are difficult to deploy in a high-performance computing environment. Additionally, the previous methods overlook the simplification of the problem that can be achieved by incorporating knowledge of the road network connectivity and direction of travel rules. This report describes a matching algorithm that was developed to avoid computational challenge and reduce errors through incorporation of road network connectivity.

CHAPTER 2. THE ALGORITHM

The goal of the matching algorithm is to computationally reconcile the latitude/longitude path from the GPS with the digital map data to efficiently and accurately identify the roads that were traveled. This section will describe the method used to do the matching in general terms, including the general data sets and infrastructure. The next section of the report will provide more detail on the logic employed by the matching algorithm.

The translation of GPS records into a list of roads traveled can be considered a flow of data from the source devices through storage servers and computational environments into a summary database. A flowchart depicting this sequence is provided in Figure 1.

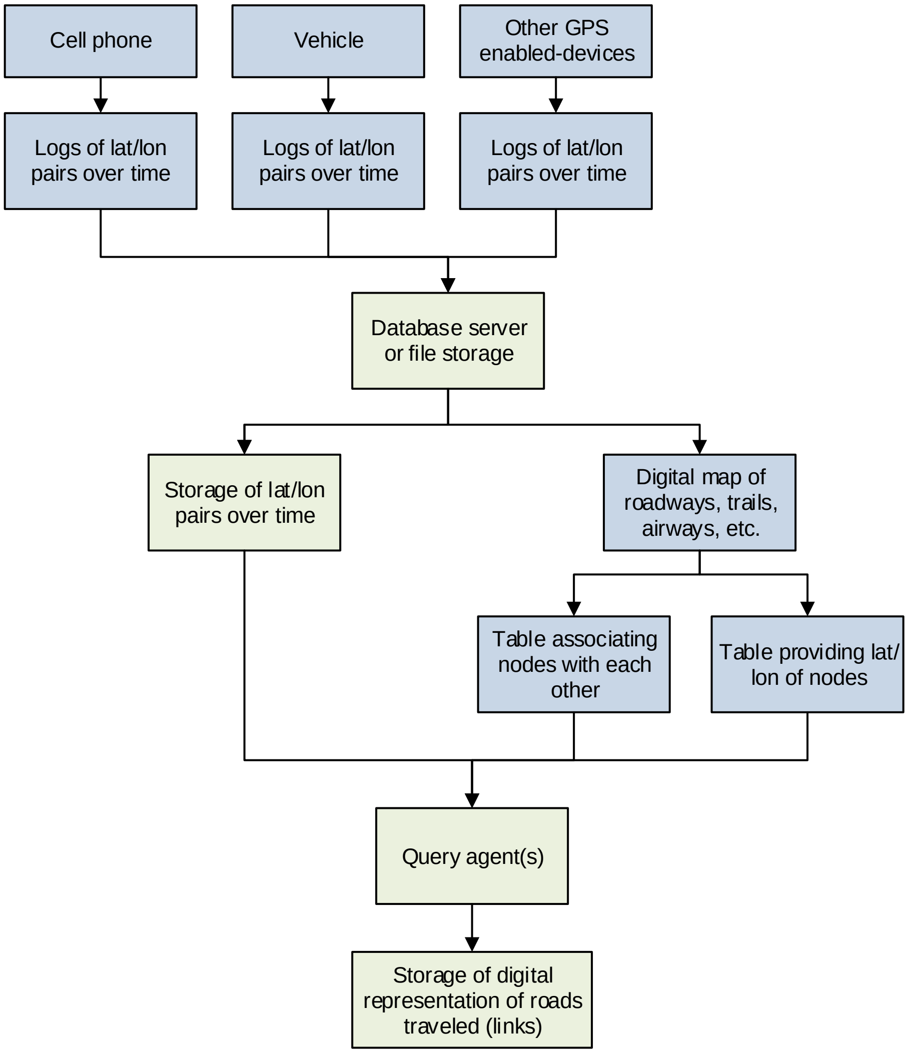
Cell phone
Vehicle
Other GPS enabled-devices
Logs of lat/lon pairs over time
Logs of lat/lon pairs over time
Logs of lat/lon pairs over time
Database server or file storage
Storage of lat/lon pairs over time
Digital map of roadways, trails, airways, etc.
Table associating nodes with each other
Table providing lat/lon of nodes
Query agent(s)
Storage of digital representation of roads traveled (links)

Figure 1. Flowchart. Components in the translation from device data into a database describing routes.

Logs from the GPS-enabled devices (e.g., cell phone, vehicle system, data acquisition system) are stored on a database server or file storage. These logs include latitude, longitude, timestamp, and dilution of precision values. These values, among others, are standard on essentially any GPS device.

Digital map data can be hosted on the same server or another server. Digital map data are representations of travel networks, including roads, paths, trails, airways, etc. They generally include links that join at nodes representing intersections. Figure 2 illustrates digital map links (maroon) and nodes (black).

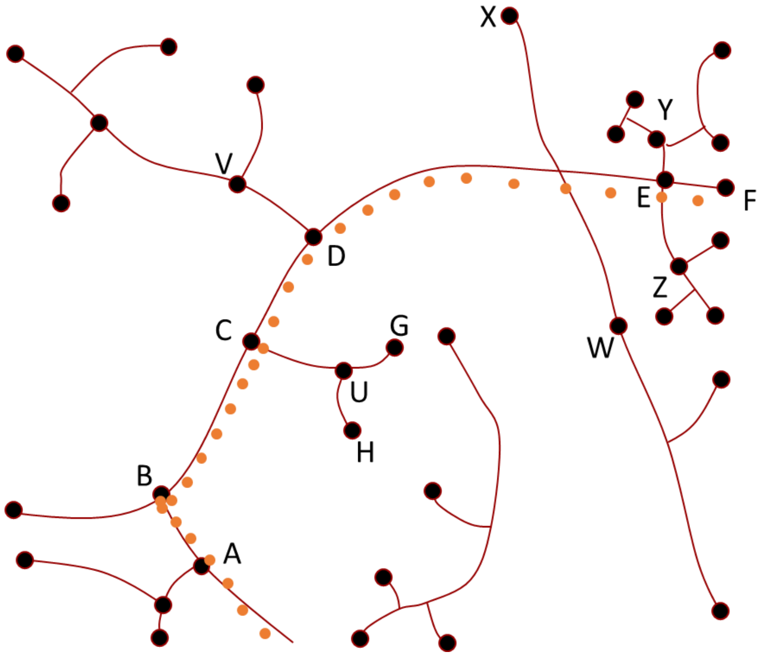

Figure 2. Diagram. Representation of GPS log data overlaid on digital map data.

Figure 2 also shows GPS points (in orange) overlaid on the road network. Both the latitude/longitude path from the GPS and the digital map data include error. The GPS records include error between the actual position of the device and the reported position. Digital maps include measurement error from measurement, registration, etc. In Figure 2, the separation between GPS points and the road location could be due to either a poor solution of position by the GPS or the digital map not locating the road in the exactly correct location.

In this example, visually it is straightforward to identify the links that were traveled. The goal of the matching algorithm is to identify the links, AB, BC, CD, etc., through an automated process. Note that in some places a GPS point appears to be on a link that was not actually part of the route (e.g., CU and EZ). The algorithm should not falsely identify these links as part of the route.

A critical element of the matching algorithm's implementation is that the map database includes tables defining how nodes relate to each other. For example, a map database describing the network in Figure 2 would indicate that from Node C, Nodes D, U, and B can be reached directly but not Nodes V, A, etc.

Both the map and GPS logs are stored in relational database tables, which permits the processing of both types of data using standard database tools. This bypasses graphical interfaces and related file formats that require large computing resources for processing and rendering. Common database batch processing tools, database server clusters, and computational clusters can be used.

To process the two data sources into a route, query agents, such as a database package on a desktop client or a customized program running on a computational cluster, connect to the database servers and access both the GPS logs and the digital map data. As the agents operate on the input data, the matching algorithm reconciles the GPS data with the digital map data, identifies the actual roads, and stores the record of the route of travel for subsequent use. The next section provides details on how this reconciliation is accomplished.

DETAILS

The components of the matching algorithm are described in the flowchart in Figure 3.

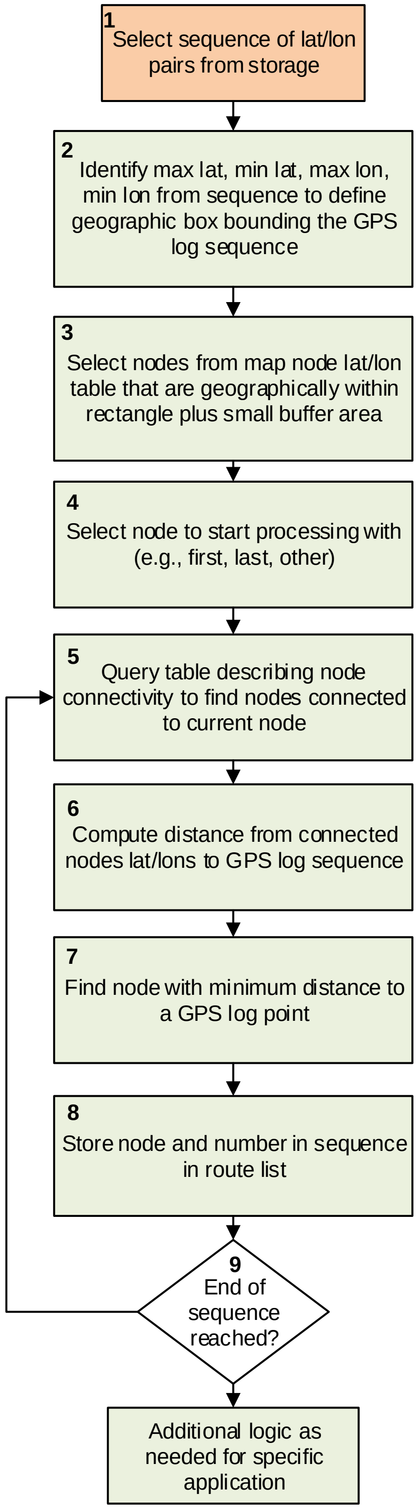
1
Select sequence of lat/lon pairs from storage
2
Identify max lat, min lat, max lon, min lon from sequence to define geographic box bounding the GPS log sequence
3
Select nodes from map node lat/lon table that are geographically within rectangle plus small buffer area
4
Select node to start processing with (e.g., first, last, other)
5
Query table describing node connectivity to find nodes connected to current node
6
Compute distance from connected nodes lat/lons to GPS log sequence
7
Find node with minimum distance to a GPS log point
8
Store node and number in sequence in route list
9
End of sequence reached?
Additional logic as needed for specific application

Figure 3. Flowchart. Process flowchart.

Step 1. A sequence of latitude/longitude pairs is retrieved from storage. This is frequently done through query of a database, but the data could also be retrieved from files. Within the transportation context, a common retrieval would likely consist of coordinate pairs for one trip, but a smaller epoch of time could be queried.

Step 2. A geospatial bounding box is identified that includes all the points of interest. This can be done by selecting the maximum and minimum latitudes and maximum and minimum longitudes from the epoch. This step will be used to reduce the area of digital map data that is considered by the matching algorithm. This selection of a patch of map data is not necessary in all applications, but it provides a way of reducing the number of comparisons required, as well as the overhead associated with moving and storing additional data beyond what is of interest.

Step 3. The digital map data are then queried to identify nodes that exist within the bounding box identified in Step 2. An additional buffer of area (e.g., bounding box edge plus some distance) is also helpful so that nodes just beyond the retrieved points are also included in consideration during subsequent steps.

These nodes are retrieved along with their position and information describing the other nodes to which they are connected. In some databases, this connectivity is directly available. In others, it would need to be generated from commonly available "from/to node" information associated with the links.

Step 4. The next step is to find the node within the map patch that is geographically close to the GPS log latitude/longitude pairs. Logically, this is often the beginning of the temporal sequence of GPS latitude/longitude pairs, and subsequent processing would progress forward in time through the log. However, it is also reasonable to work backwards through the log, or even progress in both temporal directions. These additional approaches can be used, for example, to explore alternative solutions and compare one against the other.

Step 5. Once the first node for processing is chosen, the next step is to query the map table to identify which nodes are connected to the first node. For example, when processing Node C in Figure 2, the simplest application of this would return Nodes D, U, and B, as well as the latitude and longitude for each.

Step 6. The next step is to calculate the distances to the connected nodes. In the example of Figure 2, the distances from Nodes B, C, and D to the log data GPS points are computed.

Step 7. Of the connected nodes, the one that is found to have the minimum distance to the log points is the next node in the sequence. If it is found to be Node D, as is the case in this example, this would indicate that link CD had been traversed.

Step 8. The next step is to store the information on the traversed link in the output table, which would be data pertaining to Node D in the example.

Step 9. The final step determines if the end of the log sequence has been reached. If it has not, the process returns to Step 5, querying what nodes are connected to Node D.

The simple retrieval of proximate nodes described in Step 5 can be made more robust by retrieving more distal nodes in the network. For example, if a solution has arrived at Node B, the algorithm can query the network to look two or three links ahead in every possible direction. Route alternatives retrieved in this step would include BCUG, BCUH, and BCDV, as well as routes past Node T (BT_ _, etc.). In this application, Step 7 would check the distance of the vehicle-reported GPS points to each of the nodes in the sequence, and the correct sequence could be selected through logic, such as the minimization of the average error between the vehicle latitude/longitude and the node latitude/longitude at all of the nodes along a route alternative. This approach was used in this work to choose the path of travel. This reduced the potential for a large error at an individual node amidst other nearby nodes.

Once the end has been reached, additional logic can be applied as needed for specific applications. This logic may use additional data to validate the output. For example, heading information from the GPS logs might be compared to node position data or link travel direction data to confirm that the selected path of travel is realistic.

NATURALISTIC DATA APPLICATION

The methods described here have been applied to four naturalistic data sets, most notably the SHRP 2 data set, which will be summarized here. The SHRP 2 GPS data were recorded in approximately 5.5 million trips collected for 3,147 drivers at six sites in the continental United States. This represents approximately 3.7 billion latitude/longitude pairs which needed to be matched with roads. A NAVTEQ 2012Q2 digital map was used. The time period covered in this release coincided with the core collection of driving data in the study. This digital map includes over 40 million links, which represent the North American Road network. The map data are stored as a relational table structure with, for example, one table listing links and associated attributes and another table listing nodes and their attributes.

During processing, the GPS data were left in place within the database tables making up the SHRP 2 data collection. These data are stored on an IBM DB2 database cluster with one head node and four processing nodes. The map data were also stored on these servers to facilitate access during processing.

The matching algorithm was coded as a set of functions in MATLAB. Functions handled tasks such as retrieving GPS and map data, developing a set of route alternatives, evaluating the route alternatives against the GPS data, writing the results to the database, and handling process tracking, surrounding batch processing, and computational cluster implementation tasks.

The algorithm was deployed on a computational cluster with varying numbers of computational nodes available, ranging from 32 to 96. When operating at maximum throughput, the algorithm and infrastructure could read in data, process the data within the matching algorithm, and write a solution to the database for approximately 29 trips per second. Phrased another way, every second approximately 22,000 latitude/longitude pairs were matched to links from within 40 million links in North America and stored as routes along the road network.

The output of the process for the SHRP 2 data set generated over 305 million rows of data, with each row representing a link defined in the digital map. In addition to the number identifying the link, the output table identifies the file and provides the file timestamp at the start and end of the link, and a measure of the distance between the reported GPS position and the digital map node at the start of the link. This final stored measure provides a straightforward method for researchers to evaluate the accuracy of the solution. If the distance between the vehicle's reported position and the map becomes large (e.g., greater than 40 feet), the solution may not be accurate.

This table can be queried in seconds for common research tasks. For example, a query can count the number of traversals available in the data set on a link of interest. A query identifying the most commonly traveled link in the SHRP 2 data set indicates a link with 24,441 traversals by 672 different participants. The route data can also be aggregated by link and presented as a map indicating the frequency of travel on different links.

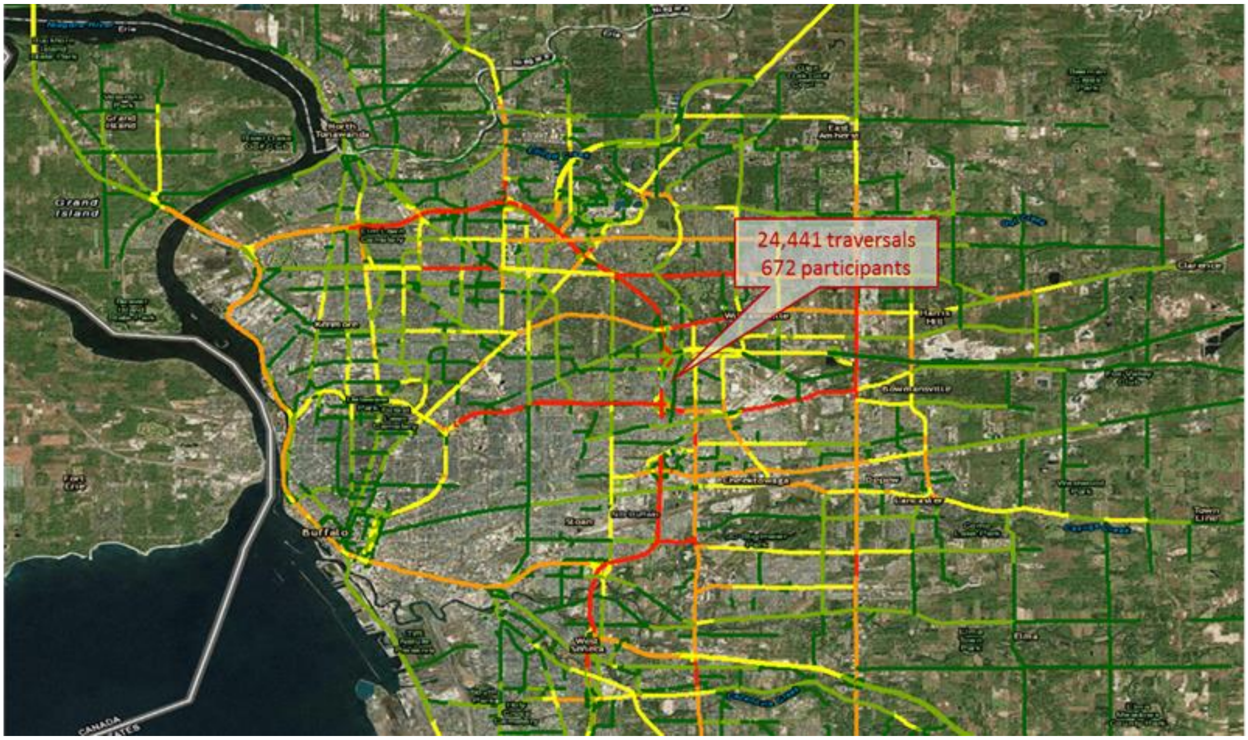

Figure 4. Map. Map characterizing link counts in the Buffalo, NY, area (red indicating many, green indicating few).

The numeric link identification number can be joined with other data sources to identify roadway attributes such as the names of roads, number of lanes, speed limits, road classes, etc.

VALIDATION

The output of the matching algorithm and surrounding process were evaluated against manual route identification results. The first step in this work was to open 100 randomly selected SHRP

2 trips in the Virginia Tech Transportation Institute's (VTTI's) data viewer and overlay the GPS data on a map. A data reductionist then typed the road names that the participant had traveled into an Excel spreadsheet. The reductionist also recorded the timestamp at the beginning and end of each road.

Separately, MATLAB code was developed that would translate the matching algorithm output for these 100 trips into a sequence of street names and timestamps. Because the digital maps break streets into multiple links, the code also collapsed the roads with the same name into one segment. This output was written to the same spreadsheet to the right of the reductionist's manual list.

A separate reductionist then reviewed a map to compare the route list developed manually against the route list generated by the matching algorithm. In some cases, the reductionist made errors, such as missing subtle locations where street names changed, or including a highway off-ramp as part of a highway. In other places, the algorithm made errors, such as identifying the wrong link. Once the two were evaluated and possibly compared again against the map, the correct route solution was determined.

The algorithm was then compared against this correct solution, and a confusion matrix was created that quantified how much driving time was correctly assigned to a link, correctly not assigned to a link (e.g., if in a parking lot), assigned to the wrong link, or assigned to a link when not really on a link. This confusion matrix is provided in Table 1.

Table 1. Validation confusion matrix.

		Algorithm				
		Link N	Not on Link			
Actual	Link N	6,470,066 sec	620,613 sec	91%	Sensitivity	Method finds x% of link time correctly.
	Not on Link	131,012 sec	774,447 sec	86%	Specificity	x% correct ignoring time that is not on a link.
		98%	56%			
		Positive Predictive Value	Negative Predictive Value			

The results indicate that the algorithm assigned driving data to the correct link 91% of the time. When the driving data were not on a link, the algorithm correctly recognized this 86% of the driving time.

CHAPTER 3. CONCLUSIONS

The procedures developed in this work are valuable for many applications because they bypass traditional geospatial tools and file formats, which have accuracy challenges when matching at any scale, and are infeasible when working at large scale. The method described here directly accesses both digital map data and GPS logs using common database tools that can be deployed on large-scale computing infrastructure. At the same time, the method reduces computational complexity through consideration of roadway network connectivity.

The application to the SHRP 2 Naturalistic Driving data set demonstrates the strengths of the method; specifically, the capability of rapidly processing large numbers of GPS points and achieving a solution with a high level of accuracy. By relying on a small number of GPS-based inputs, primarily latitude and longitude, the matching process can easily be applied to other data sets or latitude/longitude records from any number of devices or applications. The processes have already been applied to naturalistic data from motorcycles, teen drivers, and the 100-Car study data. Readers interested in a more comprehensive description of the application of the matching algorithm to SHRP 2 data are referred to *S31 Task 1.7: Naturalistic Driving Study: Linking the NDS Data to the Roadway Information Database* (McLaughlin & Hankey, 2014).

REFERENCES

Cannon, B. R., McLaughlin, S. B., & Hankey, J. M. (2009). *Method for identifying rural, urban, and interstate driving in naturalistic driving data* (Report No. 09-UT-005). Blacksburg, VA: National Surface Transportation Safety Center for Excellence.

Li, Y., Gibbons, R., & Medina, A. (2014). *Feasibility for linking an adaptive lighting database with SHRP 2 naturalistic driving data* (draft report). Blacksburg, VA: National Surface Transportation Safety Center for Excellence.

McLaughlin, S., & Hankey J. (2014). *S31: Naturalistic Driving Study: Linking the NDS data to the Roadway Information Database* (Report S2-S31-RW-3). Washington, DC: National Academies of Science.

Wu, S-C, & McLaughlin, S. B. (2012, October). Creating a heatmap visualization of 150 Million GPS points on roadway maps via SAS®. Southeast SAS User Group, Durham, North Carolina.