REAL-TIME TRAVEL PATH PREDICTION USING GPS-ENABLED MOBILE PHONES

Narin Persad-Maharaj, Sean J. Barbeau, Miguel A. Labrador,
Philip L. Winters, Rafael Pérez, Nevine Labib Georggi

University of South Florida
Center for Urban Transportation Research
Department of Computer Science and Engineering
Tampa, FL 33620
(maharaj, barbeau, winters, georggi)@cutr.usf.edu
(perez, labrador)@cse.usf.edu

ABSTRACT

This paper presents an algorithm to predict an individual's real-time travel path and destination using data from a Global Positioning System (GPS) enabled mobile phone. The algorithm uses spatially-aware, geometric representations of a user's historical trip data in its comparisons thus eliminating the need for taxing calculations. This technique also promotes scalability through the use of enterprise spatial database systems. Multiple modes of transportation are supported since the algorithm does not rely on road network information. This Path Prediction algorithm can enable a variety of services such as real-time localized traffic incident reporting, location-based advertising, and live traffic management.

Keywords – location based services, LBS, global positioning systems, GPS, mobile phones, cell phones, path prediction, destination prediction, travel behavior, GIS, TRAC-IT

INTRODUCTION

With the increasing availability of low-cost GPS technology, location-based services (LBS) applications are becoming available for the average consumer. While vehicular navigation systems have become extremely popular in recent years, the next evolution of the LBS environment is a transition to GPS-enabled mobile phones, a market expected to have 315 million users worldwide by 2011 *(1), (2), (3)*. In its September 2006 report, *US Wireless Business Location-Based Services 2006-2010 Forecast,* market intelligence firm IDC anticipated that LBS will be in use within the next four years by over half of mobile phone users in the United States *(4)*. This growth in location capabilities for cell phones in the United States is largely driven by a Federal Communication Commission (FCC) e911 mandate requiring wireless carriers to locate emergency callers to within 50 to 300 meters of their geographic position *(5)*.

Several key issues should be addressed when designing location-aware software applications that deliver real-time traveler information to GPS-enabled mobile phones. First, the user experience is severely constrained due to the limited user interface. To provide an effective user experience, every effort must be made to design intelligent applications which require user input only when necessary. Reduced user input has become more important in recent months with several states passing laws that ban the use of handheld devices while driving *(6)*. Some studies have documented that the distraction of multi-tasking, and not the physical

Presented at the 15th World Congress on Intelligent Transportation Systems, New York, New York, November 16-20, 2008. Paper # 30413.

act of holding a phone is a danger to drivers. Therefore, user interaction must be minimized to avoid distracting drivers even when wireless Bluetooth™ headset devices are used *(7)*. Companies employing location-aware applications will also want to avoid such distractions for liability reasons. Therefore, a predictive mechanism is desired that will passively monitor the traveler's behavior and give alerts to travelers before they enter major areas of traffic, or before they actually begin traveling, without requiring any user input. This always-on, push method is in contrast to traditional phone-based traffic information systems such as 511, where the driver must call a phone number and retrieve traffic information from the system *(8)*. While some 511 systems allow the user to subscribe to emailed traffic alerts about particular roadways, such subscription lists require significant user effort to register for every road they might use for traveling *(9)*. Additionally, these lists are static and do not change based on the driver's real-time location or historical travel behavior. For this reason, the user is likely to either receive an overwhelming number of irrelevant alerts if they have subscribed for many roadways, or will miss critical information for atypical travel behavior on roads for which they did not subscribe.

Another consideration of traveler information systems is that if a location-based notification is given to a motorist near an incident, it may be too late for a traveler to avoid encountering areas of congestion. LBS can inform the user of an incident before they reach problem areas, if the traveler's destination is known in advance. However, traditional vehicle-based navigation devices require the user to manually input the destination for each trip, which is not practical for everyday trips where the traveler does not need or use driving directions. This predicament could be avoided altogether if the LBS is automated to predict the user's destination and present them with route hazard warnings in advance. These same principles also apply to location-based advertising. The user would want to receive advanced notice of discounts or sales when they can alter their trip plans, not as they are driving by the location where the product or services are sold. For all of the above mentioned reasons, a real-time travel path prediction component is desired that can passively monitor the traveler's behavior and alert the traveler only when information is highly relevant to their real-time position, as well as their past travel behavior.

This paper presents an algorithm that automatically predicts travel behavior, including immediate route and destination information based on the user's time-stamped path and travel history, statistics related to their real-time path, and general path distribution for all users. This Path Prediction algorithm utilizes spatial queries, such as the features and functionality standardized by the Open Geospatial Consortium, and therefore can be easily scaled using spatial database management systems *(10)*. Scalability is critical for distributed systems that service a large number of mobile users in real-time. Since the algorithm does not require road network information, Path Prediction can be used for all modes of transportation. The algorithm was tested using real and synthetic trip data to analyze common travel behavior.

RELATED WORK

Many areas of research are being pursued that focus on providing real-time information to travelers. Youngho *et al.* are researching more efficient data transportation structures to deliver multimedia rich data that include parking information, business hours and even restaurant reservations *(11)*. Tan *et al*. have developed a vehicle collision warning system which allows nearby vehicles to communicate their positions to one another so that their drivers can avoid potential accidents *(12)*. Barbeau *et al.* have created a software system for commercially-available GPS-enabled mobile phones that is able to deliver personal prompts such as "Get Ready…" and "Pull the Cord Now!" to transit riders to aid the user in exiting the vehicle at the proper location for their particular itinerary *(13)*.

Previous research in traveler destination prediction methodologies has derived a user's trip destination through many different methods. Simmons *et al.* use a Hidden Markov Model (HMM) to capture a sequence of driver actions used to navigate a route and deduce the destination based on a present and next state transition model *(14)*. Marmasse and Schmandt use histogram matching, as well as a Hidden Markov Model (HMM) and a Bayes classfier, in their *comMotion* research to surmise a user's destination *(2), (15)*. Ashbrook and Starner used Markov models trained to predict a next or most likely destination candidate based on recently visited destinations by using clusters of GPS data as inputs *(16)*. Project Lachesis focuses on a location clustering algorithm that is used to "model transitions between clustered locations using a Markov Model" *(15), (17)*. Liao *et al* takes a different approach and uses a Bayesian network hierarchy, demonstrating that it outperforms a second order Markov Model *(15), (18)*. Karbassi *et al*. developed an algorithm that uses probability histograms taken over a period of three years for discrete daily time intervals to match the path of a bus passenger to one of five fixed bus routes *(15), (19)*. Depending on the time of day, their algorithm anticipates that a bus will travel on a certain route, and in a certain direction.

Unlike the aforementioned methods, Path Prediction does not make exclusive or widespread use of probabilistic methodologies to determine a traveler's destination. Instead, it primarily utilizes spatial queries. Spatial queries are geospatial calculations performed on GIS data in the form of shapes such as multi-line segments or polygons, which could represent a trip path or land parcel. Many spatial features and functionality have been standardized under the Open Geospatial Consortium *(10)*. The use of spatial queries allows the algorithm to be easily scaled through the utilization of existing commercially available spatial database management systems such as Oracle Spatial, PostGIS, and ESRI ArcGIS products *(20), (21), (22)*. Scalability is a critically important feature when designing a distributed system that will simultaneously serve many mobile phone users in real-time. The use of spatial queries also allows the path prediction algorithm to be independent of any single mode of transportation, and could even be used for walking trips if the user traversed a regular set of paths.

PATH PREDICTION METHODOLOGY

Path Prediction utilizes an individual's travel patterns to determine his or her future travel path. If the user does not have a sufficient path history, the algorithm can transition between individual user path history and all path records that are stored in the spatial database. This principle allows for a more individualized route prediction approach that is based on the user's specific travel patterns, rather than statistics based solely on the general populace or commerce/business popularity.

In order to process the user's travel history for real-time path prediction, an application referred to as the Path Maker is responsible for cataloging the user's trip details, which includes GPS coordinates. Once these statistics and data are archived in a GIS database, the Path Prediction algorithm can then anticipate the user's travel route based on his or her past travel patterns and send any alerts related to the path. Utilizing standard spatial queries to make predictions reduces the complexity of the process, as compared to required calculations presented in previous studies also focused on path prediction *(2), (15), (16)*. For example, the implementation of spatial queries can approximate complex polygon shapes as a series of bounding boxes, thereby enhancing the scalability of the system and allowing many processes to execute simultaneously in real-time, which is critical for real-time travel information systems for mobile phones. These simpler representations of user travel should be more than sufficient for services such as targeted personalized traffic incident information, immediate traffic demand forecasting, or location-based advertising.

Path Maker Algorithm

The Path Maker is responsible for creating and maintaining an archive of user trip records (Figure 1). It queries a SQL database table for trip coordinates categorized by trip, user identification (ID), and trip time. From the trip coordinates, the application constructs a geographic and geospatially accurate polyline and makes calculations. A polyline is a set of straight line segments connected to form a larger, often curved, path. Consequently, it requires a minimum of two points to be considered a polyline.

Once the coordinates have been queried and retrieved and the polyline assembled, the Path Maker algorithm buffers the polyline with a five meter radius to create a polygon with a total width of ten meters. This polygon is then stored in a GIS server spatial database along with its statistics, such as the trip, user ID, and a timestamp that consists of the trip's start and end times. Storing polygons is important to Path Prediction because it is easier and more efficient to do spatial calculations with geometries that have a stored and indexed spatial area.

Path Prediction Algorithm

Once the GIS database is populated with polygon records, the Path Prediction algorithm (Figure 2) can then query it for records to compare against a user's real-time path. Algorithm execution is initiated by an application that is designed to continuously run in the background on a mobile phone *(23)*, *(24)*. This mobile application will begin sending GPS data to the server when movement indicates the user may be beginning to travel. GPS, or accelerometers if available, can be used as movement detection technologies. The path predictor awaits the first five GPS coordinates to be received from the phone. To ensure only useful GPS data is

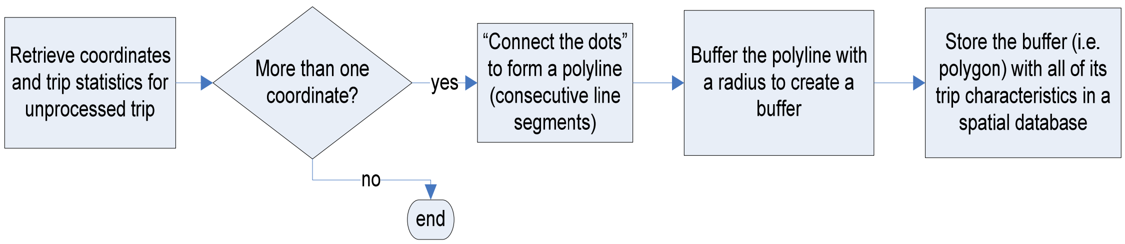

Figure 1 - Path Maker Flowchart

entered into the algorithm (i.e., GPS data that represents a real travel path and not GPS drift in a stationary position), pre-filtering techniques such as a critical point algorithm may be utilized *(25)*. The algorithm then forms a polyline using these coordinates and performs a polyline-in-buffer spatial query against the spatial database holding the historical record of trips. If multiple buffers are retrieved from the query, the algorithm refines the results by comparing the traveler's user ID to those of the polygons, and eliminates any polygons with a different user id. If multiple polygons still remain, the algorithm then uses the start location of the trip and eliminates any buffers that have different start locations.

Presented at the 15th World Congress on Intelligent Transportation Systems, New York, New York, November 16-20, 2008. Paper # 30413.

After this process is performed, the best case, but least likely scenario is that only one polygon record is returned representing the probable, immediate traveler path. The algorithm can test for an intersection of the detected polygon with any known incidents or registered location-based advertising locations, and deliver appropriate alerts to the traveler via the mobile phone. The alert can take many forms, such as text message or recorded audio alert *(26)*. If multiple polygons are detected as possible trip paths, system policies may decide whether to query the database for registered incidents or advertisements, and provide alerts to the user. Different types of system policies are discussed at the end of this section. Normally, a user will only be alerted once per accident or advertisement, so that duplicate notifications will not be sent in future iterations.

The worst case scenario is that no polygon records are returned after this initial query process. This can occur if a polygon record was never created for a previously recorded path, or if the spatial database is sparsely populated with trips. If no polygons are returned, the algorithm performs a series of steps to detect polygons in the nearby area. First, the oldest coordinate in the five coordinate set will be eliminated, and the algorithm will be called again in a recursive fashion until any polygon records are detected. This process gradually eliminates the end of the current trip in an attempt to match it to previously recorded trips that may have started at a

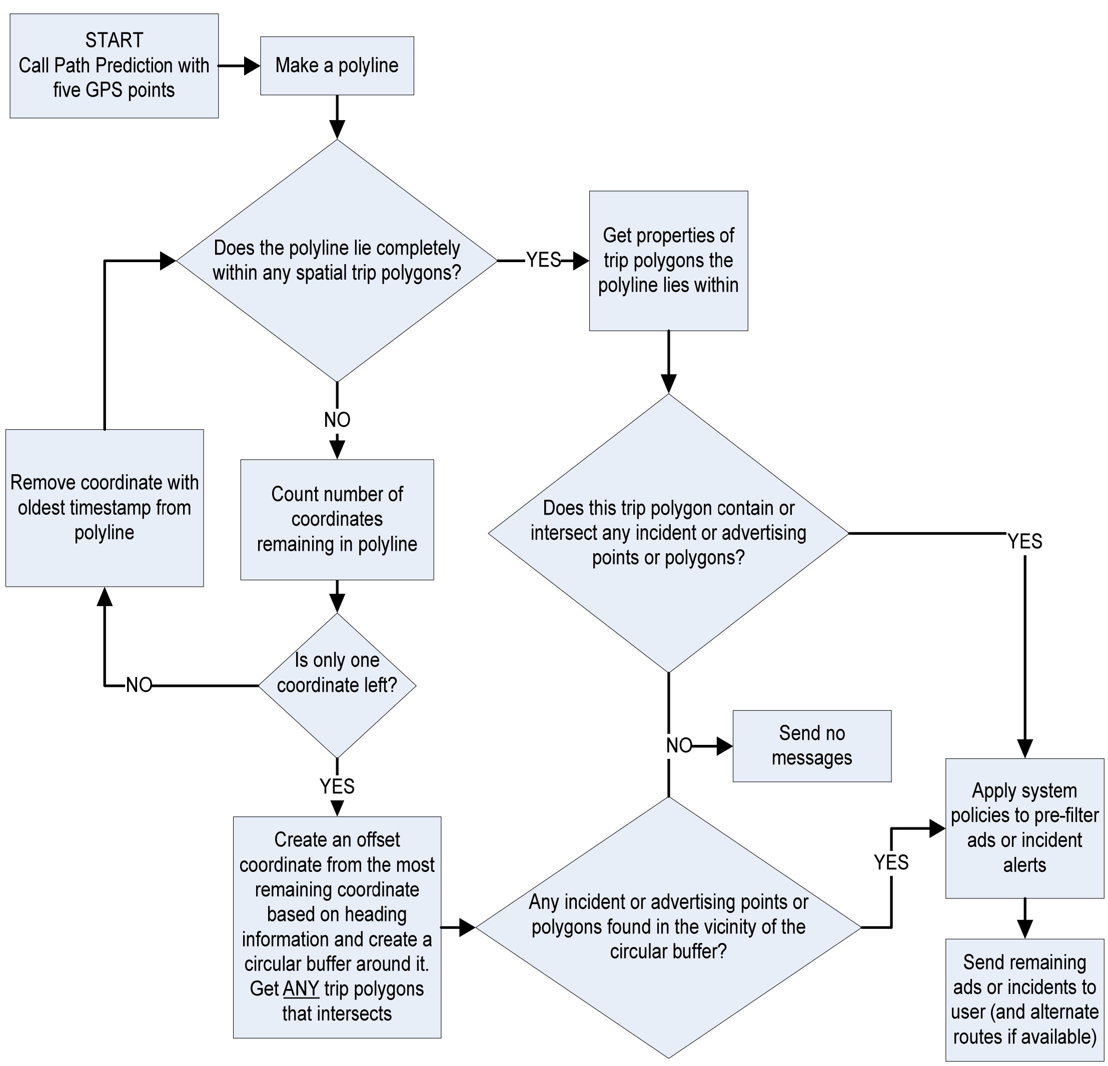

Figure 2 – Path Prediction Algorithm Flowchart

different location. These steps are performed because the success of a polyline-in-buffer spatial query requires that a polyline be completely confined within the polygon. This situation might occur if the user's current trip begins recording before his or past trips did. If only one coordinate remains from a result of the recursive method and no polygons have been found, then the heading of the most recently recorded GPS coordinate is used to project a predicted next coordinate in a final attempt to identify nearby historical trip records.

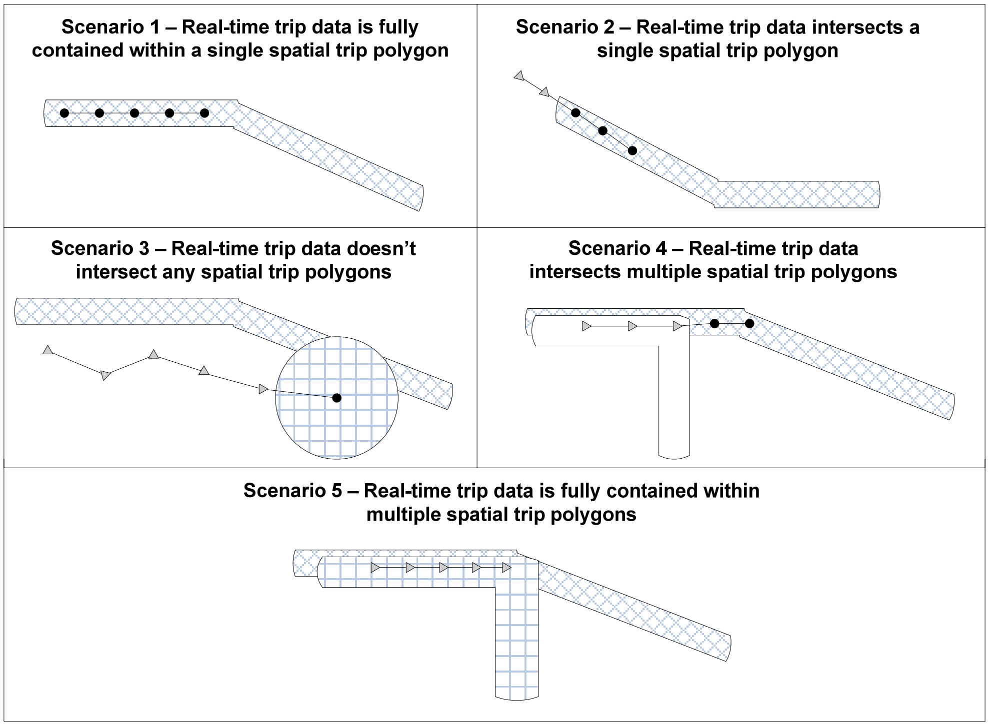

Figure 3– Sample Executions of the Path Prediction Algorithm for Five Different Possible Scenarios

A 10-meter radius is created around this point, and an intersection query against the polygon database is performed to retrieve any historical trip polygon records within the area. This projection of a possible next coordinate is performed to give the user immediate feedback about any hazards or registered alerts within their vicinity, rather than send them no information. Once the mobile phone application obtains a new set of five coordinates, the path prediction algorithm is launched again and the aforementioned procedures will be performed using this new GPS data. This process will continue until the user stops traveling so that the Path Prediction algorithm can constantly be looking ahead to warn the user of possible incidents, or alert them of location-based advertising. The trip that was just recorded is then archived in the spatial database for later batch processing by the Path Maker.

Ideally, a user's travel history will be rich enough to provide at least one trip that was recorded near their real-time location that can be used for path prediction. Many automotive trips have an initial walking period associated with the start of the trip, so it is possible to deliver alerts before a user even starts their vehicle. In order to minimize the distraction to the user while they are traveling, the system can schedule alerts to be delivered only when the user has stopped moving for an extended period of time (i.e., the user is stopped at a red light).

Different constraints on delivering alerts to the user based on their predicted path may be appropriate for the algorithm, depending on the type of notification to be sent to the traveler,

Presented at the 15th World Congress on Intelligent Transportation Systems, New York, New York, November 16-20, 2008. Paper # 30413.

and the impact of a false-positive (i.e., an alert is sent to a user when it is not relevant to that user) or a false-negative (i.e., an alert is not sent to the user when it would have been relevant to that user). This situation typically arises when multiple trip polygons are detected by the algorithm as possible paths. For example, the traveler may want to receive traffic accident alerts for all possible paths since the impact on the user for a false negative is time wasted sitting in traffic, when they could have avoided the congestion. However, the user may only want to receive highly targeted alerts for location-based advertising, since too many false positives will result in the user receiving many advertisements irrelevant to their current trip. The Path Prediction algorithm can be configured by system administrators to deliver alerts based on system policies as well as user profiles and preferences. User profiles and preferences will play a major role for location-based advertising. It is therefore critical that only advertisements that the user has subscribed to, and those that are highly relevant to the user's real-time predicted path, should be sent to that user.

Sample Execution Scenarios

Several sample algorithm executions are shown in Figure 3 to illustrate how the Path Prediction algorithm reacts in different circumstances while the user is traveling. In each of the scenarios, five critical points have been recorded by the mobile phone and have been passed as input to the Path Prediction algorithm. The direction of travel for each scenario is from left to right.

Scenario 1 – Real-time trip data is fully contained within a single polygon: Since the real-time path is fully contained within a single buffer (shaded polygon), this polygon is used for real-time path prediction. The shaded polygon is checked for intersection with any incident or advertising shapes (i.e., points, polylines, or polygons), and if there is an intersection, the user is notified.

Scenario 2 – Real-time trip data intersects a single polygon: Since the first two coordinates (denoted by triangles) did not fall within a trip polygon, they were removed one-by-one until the entire remaining polyline was completely within the trip polygon. This polygon is then used to predict the user's immediate travel path. The algorithm looks to see if any incident shapes intersect with this trip buffer, and if so, the system notifies the user.

Scenario 3 – Real-time trip data doesn't intersect any polygons: Since the first entire trip polyline (denoted by triangles) did not fall within a trip polygon (even after the oldest points are removed one-by-one), an offset coordinate (circular point at center of radius) is created based on the direction of travel of the last recorded point (the point immediately to the left of circle). A spatial radial buffer (shaded circle) is then projected around this offset coordinate. In this scenario, the radial buffer intersects with a trip polygon, and this trip buffer is used to predict the user's immediate path. The algorithm looks to see if any incident or advertising shapes intersect with this trip buffer, and if so, the system notifies the user.

Scenario 4 – Real-time trip data intersects multiple polygons: In this scenario, the real-time path intersects two buffers (white and shaded). However, since the entire polyline lies completely within only the shaded buffer (i.e., the circular points intersect only the shaded buffer), the shaded polygon is used to predict the user's immediate path. The algorithm looks to see if any incident or advertising shapes intersect with this shaded trip buffer, and if so, the system notifies the user.

Scenario 5 – Real-time trip data is fully contained within multiple polygons: Since the real-time path is fully contained within two buffers, spatial data alone is not sufficient to narrow down which path user is more likely to travel. In this case, schedule data from trip polygons, such as trip day of week or time of day may be used. Probabilistic methods can also be used to determine the user's most likely path. For incident detection, system policies that weigh the benefits of the type of notification, and the impact on the user for false negative and positive identification, are considered before deciding whether to use all, or a subset of returned polygons for incident or advertising shape intersection and user notification. For example, in traffic incident notifications it is often preferable to scan all of the possible paths simultaneously for accidents, since the user will want to receive information for an incident that they may encounter.

CASE STUDY

For initial implementation and proof of concept testing, a family of GIS products produced by ESRI was utilized *(18)*. The path maker and path prediction were implemented using ESRI's Java ArcObjects API, as well as ESRI's SDE spatial database system to store spatial data. The Microsoft SQL Server database management system was used to store path statistics and coordinates. ESRI's ArcMap software was used to visualize and analyze the trip buffers.

The following simulation was performed using GPS data recorded by a Sanyo 7050 GPS-enabled mobile phone on the Sprint-Nextel CDMA network, and using the TRAC-IT Java Micro Edition (Java ME) mobile phone software previously developed by the research team *(24)*. A pre-recorded trip, Trip A, was used as simulation of the traveler's real-time trip recorded on September 6, 2007. Trip buffers prior to September 6, 2007 were processed by the Path Maker and stored in the spatial database for path detection. By using Trip A, the known destination of the trip can be compared against algorithm output to determine how long the algorithm would have taken to return a single buffer that encompassed the end of Trip A. It should be noted that the algorithm can take action as early as the first iteration of the algorithm and does not necessarily have to wait for a single correct buffer to be returned before alerting the user. The amount of time spent to detect a single buffer serves mostly as an informative benchmarking tool for the algorithm and a means to compare algorithm performance on different trip instances. The simulation took 29 seconds, or 56 iterations, until a single buffer was returned by the algorithm that intersected the known destination for Trip A (Figure 8). The current estimate for the server to receive a single coordinate from the phone is four seconds. Therefore, in a real-time test, the algorithm would have taken around three minutes to identify a single buffer containing the known destination. For simplicity and clarity, only five selected iterations (Figures 4, 5, 6, 7, and 8) are shown here since much iteration returned the same detected buffers.

DISCUSSION

The algorithm functioned as expected and was able to detect multiple historical trip buffers that represented the user's future travel path. In the first iteration, only a few buffers were initially detected because the user started the trip away from any main streets. Buffers that were detected in future iterations could have been trips that started at the same location as those detected in the first iteration, but a delay in the mobile phone acquiring an initial GPS fix can result in GPS data not being recorded at the beginning of the trip. As the user moved on to a major road, the number of detected buffers doubled. GPS inaccuracies, also referred to as GPS drift, contribute to some buffers being missed even though they have the same basic shape as those being detected by the spatial query. This can be seen in trip buffers 1645 (Figure 4), and 1882 and 1884 (Figure 5). Buffers 1882 and 1884 (Figure 5) are not detected in the first iteration since the real-time GPS path differed slightly from the beginning of the

1882 and 1884 trips. However, it is very likely that the user actually traveled on the exact same road for each of these trips. Future work will focus on the ability to combine buffers with similar spatial features to avoid missing certain trips that could provide useful information to the traveler.

Figure 4 – Iteration 1: Two Paths Predicting the User's Destination are Detected

Figure 6: Iteration 40

Figure 5: Iteration 33 – The Number of Detected Paths Doubles on Major Roads

Figure 7: Iteration 51 – No Trips Detected on This Iteration

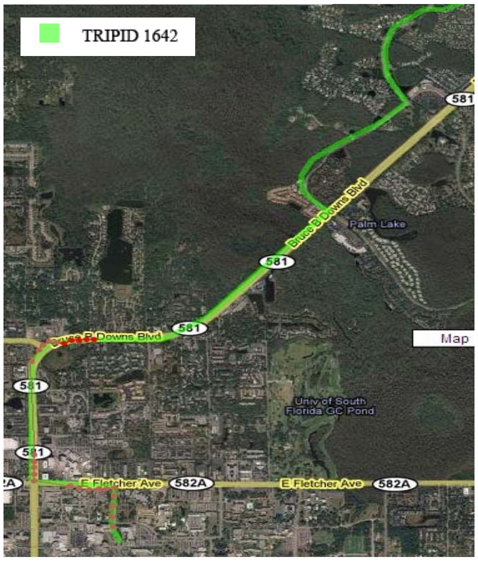

Figure 8: Iteration 56 – Known Correct Final Destination Detected in Single Trip Buffer 1642

The dynamic nature of GPS data also result in unusually large buffer configurations. In comparing buffer 1587 to its surrounding streets, it is evident that it is not perfectly aligned with the underlying street. This is attributed to GPS dropout, in which the mobile phone temporarily loses a GPS fix, but then reestablishes the fix and awareness of its location at some point later in the trip. Future work will focus on cleaning these types of buffers from the spatial database to prevent unexpected algorithm behavior.

It is important to note that it is not necessarily critical to detect a single buffer that contains the driver's intended destination when providing services such as traffic incident notification. While the trip buffers detected in the first algorithm iteration (Figure 4 - IDs 1587 and 1645) did not have the exact same destination as the final single buffer detected in iteration 56 (Figure 8 - ID 1642), much of the path area for buffers 1587 and 1645 overlapped with buffer 1642. Therefore, an incident alert triggered by an intersection with buffers 1587 and 1645 is still as useful as one triggered by 1642. Additionally, trip 1642 was actually detected in earlier iterations (e.g., iteration 40 in Figure 6), and therefore alerts for incidents intersecting that exact buffer could have been given much earlier than iteration 56. This characteristic is important for real-time performance, since relevant incident alerts can be provided to the user as early as the first algorithm iteration. Therefore, as noted earlier, system policies must carefully weigh the type of information to be delivered and the cost to the user for false-positive or false-negative detections with the number of different buffers used to trigger location-based alerts. These policies can be supplemented by user profiles and preferences to ensure that all data delivered to the user is relevant and helpful. It should also be noted that while this paper focuses on an architecture in which the Path Prediction algorithm executes on a server, the algorithm could also be adapted to execute on board a mobile device that supports programmatic spatial queries exposed via application programming interfaces (APIs). Basic spatial features, such as polygons and a method to query whether a points lies within the polygon, will be available in version 2.0 of the Java ME Location API for mobile phones *(27), (28)*.

CONCLUSION

Predictive capabilities are important for next-generation, advanced LBS on GPS-enabled mobile phones that passively monitor user location and provide targeted, relevant information to the traveler at the appropriate place and time. Prediction of the travel path also enables the system to give incident or advertising information to the user before they enter the area which triggered the alert. This gives the user more time to make informed decisions early in the trip. This paper presents a Path Prediction algorithm that uses highly scalable and configurable spatial queries to predict the user's immediate travel path based on his or her personal travel history and real-time position. The results obtained from the Path Prediction algorithm tests successfully demonstrate the proof-of-concept of the algorithm, and illustrate that the method produces the anticipated and desired results.

Future work will concentrate on enhancing the quality of the polygon network by removing or correcting GPS anomalies, increasing the width of the stored buffers to accommodate for GPS drift, and examining probabilistic techniques to further narrow down possible paths where either no buffers or multiple buffers are returned by the algorithm.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under grant No. 0453463, the Florida Department of Transportation, and the United States Department of Transportation through the National Center for Transit Research under grant numbers BD-549-24 “Testing the Impact of Personalized Feedback on Household Travel Behavior (TRAC-IT Phase 2)” and BD-549-35 “Smart Phone Application to Influence Travel Behavior (TRAC-IT Phase 3)”.

REFERENCES

(1) ABI Research. “GPS-enabled Location-Based Services (LBS) Subscribers Will Total 315 Million in Five Years,” New York, September 27, 2006 http://www.abiresearch.com/abiprdisplay.jsp?pressid=731, July 10, 2008.

(2) N. Marmasse and C. Schmandt, “A User-Centered Location Model”. *Personal and Ubiquitous Computing*, 2002 (6): p. 318-321.

(3) Rick DeMeis,. “Personal navigation devices ‘explode’ in the auto market.” EETimes July 31, 2007 http://www.eetimes.com/news/latest/showArticle.jhtml?articleID=199904600, July 2008.

(4) Schreiner, Keri, "Where We At? Mobile Phones Bring GPS to the Masses," *Computer Graphics and Applications, IEEE* , vol.27, no.3, pp.6-11, May-June 2007.

(5) Federal Communications Commision (FCC). “Enhanced 911 – Wireless Services.” October 8, 2006, http://www.fcc.gov/pshs/services/911-services/, accessed July 10, 2008.

(6) Manuel Valdes (Associated Press), “New Washington law bans hand-held phones while driving ”, http://news.yahoo.com/s/ap/20080629/ap_on_hi_te/cell_phone_driving, July 2008.

(7) Insurance Institute for Highway Safety, “1st Evidence of Effects of Cell Phone Use on Injury Crashes: Crash Risk is Four Times Higher When Driver is Using Hand-held Cell Phone,” July 12, 2005. http://www.iihs.org/research/topics/cell_phones.html, July 2008.

(8) “511 Travel Information Telephone Services”. http://www.ops.fhwa.dot.gov/511/, July 2008.

(9) “511 Tampa Bay”. http://www.511tampabay.com/Controller?appname=my511tampa&workflowname=my511&metroid=22, accessed July 10, 2008.

(10) Open Geospatial Consoritum. http://www.opengeospatial.org/, accessed July 10, 2008.

(11) Youngho Jeong, Sammo Cho, Geon Kim, Chunghyun Ahn, Sooin Lee, Whanwoo Kim, "Location based service based on digital multimedia broadcasting," *Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers. International Conference on*, January 2006, pp. 147-148, 7-11.

(12) Tan, H.-S.; Huang, J., "DGPS-Based Vehicle-to-Vehicle Cooperative Collision Warning: Engineering Feasibility Viewpoints," *Intelligent Transportation Systems, IEEE Transactions* , vol.7, no.4, Dec. 2006, pp.415-428.

(13) Sean J. Barbeau, Miguel A. Labrador, Philip L. Winters, Rafael Perez, Nevine Labib Georggi. “The Travel Assistant Device: Utilizing GPS-Enabled Mobile Phones to Aid Transit Riders with Special Needs,” 15th World Congress on Intelligent Transportation Systems, New York, New York, November 16-20, 2008.

(14) R. Simmons, B. Browning, Zhang Yilu, V. Sadekar, "Learning to Predict Driver Route and Destination Intent," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, 2006, pp.127-132,

(15) John Krumm and Eric Horvitz (2006) "Predestination: Inferring Destinations from Partial Trajectories", *UbiComp 2006: The Eighth International Conference on Ubiquitous Computing*, September 2006.

(16) D. Ashbrook and T. Starner, "Using GPS To Learn Significant Locations and Predict Movement Across Multiple Users." *Personal and Ubiquitous Computing*, 2003. 7(5): p. 275-286.

(17) R. Hariharan and K. Toyama."Project Lachesis: Parsing and Modeling Location Histories". *Geographic Information Science. Volume 3234*, January 11, 2005, p. 106-12.

(18) L. Liao, D. Fox, and H. Kautz. "Learning and Inferring Transportation Routines. in Proceedings of the 19th National Conference on Artificial Intelligence" *(AAAI 2004)*. 2004. San Jose, CA, 2004, p 348-354.

(19) Karbassi and M. Barth, "Vehicle route prediction and time of arrival estimation techniques for improved transportation system management," *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE* , vol., no., pp. 511-516, 9-11 June 2003.

(20) Oracle Spatial. http://www.oracle.com/technology/products/spatial/index.html, July 2008.

(21) PosGIS - http://postgis.refractions.net/, accessed July 10, 2008.

(22) ESRI Developer Network: http://edn.esri.com, accessed July 10, 2008.

(23) P. Winters, S. Barbeau, and N. Georggi. "Testing the Impact of Personalized Feedback on Household Travel Behavior (TRAC-IT Phase 2)" Technical Report, National Center for Transit Research, Center for Urban Transportation Research, University of South Florida. February 2008. Florida Department of Transportation. http://www.nctr.usf.edu/abstracts/abs77609.htm, July 10, 2008.

(24) P. Winters, S. Barbeau, and N. Georggi. "Smart Phone Application to Influence Travel Behavior (TRAC-IT Phase 3)" Technical Report, National Center for Transit Research, Center for Urban Transportation Research, University of South Florida. February 2008. Florida Department of Transportation. http://www.nctr.usf.edu/abstracts/abs77709.htm, June 30, 2008.

(25) Sean J. Barbeau, Miguel A. Labrador, Alfredo Perez, Philip Winters, Nevine Georggi, David Aguilar, Rafael Perez. "Dynamic Management of Real-Time Location Data on GPS-enabled Mobile Phones," UBICOMM 2008 – The Second International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies, Valencia, Spain, September 29 – October 4, 2008.

(26) Sean J. Barbeau, Miguel A. Labrador, Philip Winters, Rafael Perez and Nevine Labib Georggi, "A General Architecture in Support of Interactive, Multimedia, Location-based Mobile Applications", IEEE Communications Magazine, Vol. 44, No. 11, pp. 156-163, November 2006.

(27) Sean J. Barbeau, Miguel A. Labrador, Philip L. Winters, Rafael Pérez, Nevine L. Georggi, Location API 2.0 for J2ME – A New Standard in Location for Java-enabled Mobile Phones, Computer Communications, Volume 31, Issue 6, pp. 1091-1103, 18 April 2008. doi:10.1016/j.comcom.2008.01.045.

(28) Sun Microsystems, Inc. Java Specification Request 293: Location API 2.0. http://jcp.org/en/jsr/detail?id=293, accessed July 10, 2008.